

# MacTech®

*The Journal of Macintosh Technology and Development*

## Putting An MP3 PLAYER In The Dashboard

*by Ron Davis*

\$8.95 US  
\$12.95 Canada  
ISSN 1067-8360  
Printed in U.S.A.



# LA to New York

(and Seattle and Miami)

in 150 milliseconds or less\*

▼ Edit View Special

Your Application ⌘N

Multi-point Meetings

Real-time Collaboration

Virtual Whiteboard™

Instant Assessment™

Best-quality video

Multimedia Presentation



Call for a TeraMedia® demo

**TeraGlobal**  
COMMUNICATIONS

(858) 404.5500 ext 5660  
[www.teraglobal.com](http://www.teraglobal.com)  
[sales@teraglobal.com](mailto:sales@teraglobal.com)



Interact. Collaborate. It's highest-quality multi-point video conferencing and true real-time multimedia collaboration. It's all done in software. Any number of users. Over any distance. At any time. In real time. It flies, you don't.

\* Actual video latency varies with bandwidth. In most cases, TeraMedia® latency is below the threshold of human perception.

**TeraMedia®**  
by TeraGlobal

*"Never forget that the most powerful  
force on earth is love."*

—Nelson A. Rockefeller

Move over love.



Get a new Apple PowerMac™ G4  
or one of our refurbished models  
at price you will L-O-V-E.



**Small Dog  
Electronics**

1673 Main Street  
Waitsfield, VT 05673 USA  
Phone: **802-496-7171**  
Online: **smalldog.com**



Apple Specialist

## How To Communicate With Us

In this electronic age, the art of communication has become both easier and more complicated. Is it any surprise that we prefer **e-mail**?

If you have any questions, feel free to call us at 805/494-9797 or fax us at 805/494-9798.

If you would like a subscription or need customer service, feel free to contact Developer Depot Customer Service at 877-MACTECH

### DEPARTMENTS

**Orders, Circulation, & Customer Service**

**Press Releases**

**Ad Sales**

**Editorial**

**Programmer's Challenge**

**Online Support**

**Accounting**

**Marketing**

**General**

**Web Site (articles, info, URLs and more...)**

### E-Mail/URL

cust\_service@mactech.com

press\_releases@mactech.com

ad\_sales@mactech.com

editorial@mactech.com

prog\_challenge@mactech.com

online@mactech.com

accounting@mactech.com

marketing@mactech.com

info@mactech.com

http://www.mactech.com

### The MacTech Editorial Staff

**Publisher** • Neil Ticktin

**Managing Editor** • Jessica Stubblefield

**Online Editor** • Jeff Clites

### Regular Columnists

**Getting Started – Networking**

by John C. Welch

**Programmer's Challenge**

by Bob Boonstra

**MacTech Online**

by Jeff Clites

**From the Factory Floor**

by Metrowerks

### Regular Contributors

Vicki Brown, Andrew Stone, Tim Monroe, Erick Tejkowski, Kas Thomas, Will Porter, Paul E. Seving, Tom Djajadiningrat, and Jordan Dea-Mattson

### MacTech's Board of Advisors

Dave Mark, Jordan Dea-Mattson, Jim Straus, Jon Wiederspan, and Eric Gundrum

### MacTech's Contributing Editors

- Jim Black
- Michael Brian Bentley
- TanteK Çelik, Microsoft Corporation
- Marshall Clow
- John. C. Daub
- Tom Djajadiningrat
- Bill Doerfeld, Blueworld
- Andrew S. Downs
- Richard V. Ford, Packeteer
- Gordon Garb, Cobalt Networks
- John Hanay, Apple Computer
- Lorca Hanns, San Francisco State University
- Ilene Hoffman
- Chris Kilbourn, Digital Forest
- Scott Knaster, Microsoft
- Mark Kriegsman, Clearway Technologies
- Peter N. Lewis, Stairways Software
- Bill McGlasson, Apple Computer
- Rich Morin
- Terje Norderhaug, Media\*Design-In-Progress
- Nathan Nunn, Purity Software
- John O'Fallon, Maxum Development
- Alan Oppenheimer, Open Door Networks
- Avi Rappoport, Search Tools Consulting
- Ty Shipman, Kagi
- Chuck Shotton, BIAP Systems
- Cal Simone, Main Event Software
- Steve Sisak, Codewell Corporation
- Dori Smith
- Andrew C. Stone, www.stone.com
- Michael Swan, Neon Software
- Chuck Von Rospach, Plaidworks
- Bill von Hagen
- Eric Zelenka

### Xplain Corporation Staff

**Chief Executive Officer** • Neil Ticktin

**President** • Andrea J. Sniderman

**Controller** • Michael Friedman

**Production Manager** • Jessica Stubblefield

**Production/Layout** • W2 Graphics

**Marketing Managers**

Nick DeMello and Alyse Yourist

**Events Manager** • Susan M. Worley

**Network Administrator** • Ben Baumer

**Accounting** • Jan Webber, Marcie Moriarty

**Customer Relations** • Laura Lane

Susan Pomrantz

**Shipping/Receiving** • Joel Licardie

**Board of Advisors** • Steven Geller, Blake Park, and Alan Carsrud

All contents are Copyright 1984-2001 by Xplain Corporation. All rights reserved. MacTech and Developer Depot are registered trademarks of Xplain Corporation. RadGad, Useful Gifts and Gadgets, Xplain, DevDepot, Depot, The Depot, Depot Store, Video Depot, Movie Depot, Palm Depot, Game Depot, Flashlight Depot, Explain It, MacDev-1, THINK Reference, NetProfessional, NetProLive, JavaTech, WebTech, BeTech, LinuxTech, MacTech Central and the MacTutorMan are trademarks or service marks of Xplain Corporation. Sprocket is a registered trademark of eSprocket Corporation. Other trademarks and copyrights appearing in this printing or software remain the property of their respective holders.



This publication is printed on paper with recycled content.

**MacTech Magazine** (ISSN: 1067-8360 / USPS: 010-227) is published monthly by Xplain Corporation, 850-P Hampshire Road, Westlake Village, CA 91361-2800. Voice: 805/494-9797, FAX: 805/494-9798. Domestic subscription rates are \$47.00 per year. Canadian subscriptions are \$59.00 per year. All other international subscriptions are \$97.00 per year. Domestic source code disk subscriptions are \$77 per year. All international disk subscriptions are \$97.00 a year. Please remit in U.S. funds only. Periodical postage is paid at Thousand Oaks, CA and at additional mailing office.

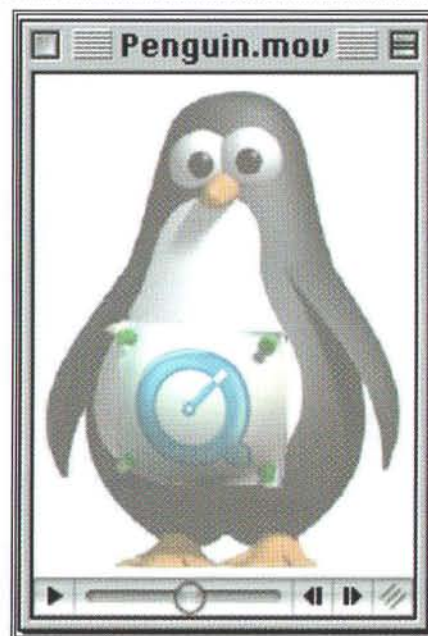
**POSTMASTER:** Send address changes to **MacTech Magazine**, P.O. Box 5200, Westlake Village, CA 91359-5200.

# C o n t e n t s

March 2001 • Volume 17, Issue 03

## Feature Articles

- 30** **QUICKTIME TOOLKIT**  
**A Goofy Movie**  
*by Tim Monroe*
- 58** **ADC DIRECT**  
**Carbon: An Essential Element of Mac OS X**



QuickTime Toolkit.....Page 30

- VIEWPOINT**
- 4** **PalmSource 2000**  
A Report from the Palm Developers Conference  
*by Marshall Clow*

- PROGRAMMER'S CHALLENGE**
- 12** **DragSort**  
*by Bob Boonstra*

- MACTECH ONLINE**
- 74** **Smalltalk**  
*by Jeff Clites*

- PROGRAMMING TECHNIQUES**
- 7** **Chart Generation with AppleScript and Excel**  
*by Rich Allen*  
*Edited by Cal Simone*

- COVER STORY**
- 62** **Putting an MP3 Player in the Dashboard**  
*by Ron Davis*

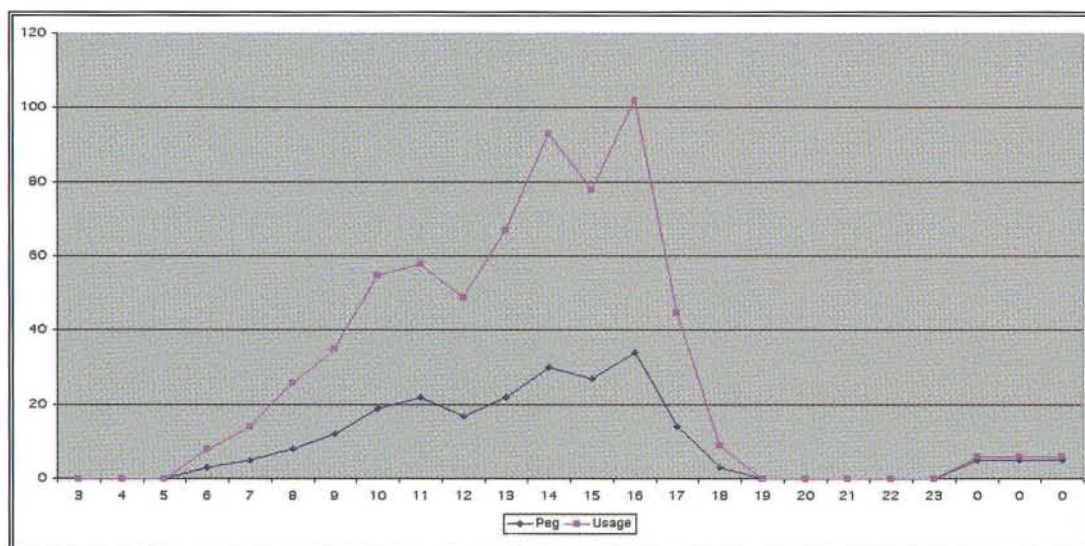


Chart Generation with AppleScript and Excel.....Page 7

By the Marshall Clow

# PalmSource 2000

## Report from the Palm Developers' conference

### INTRODUCTION

PalmSource 2000 was held at the Santa Clara Convention Center on December 12th-15th, 2000. There were over 3500 developers in attendance, up from about 2000 in 1999. As a result of this growth, Palm announced that starting in 2001, PalmSource would be held in the (much larger) San Jose convention center.

Carl Yankowski, CEO of Palm, Inc. gave the opening keynote, setting the tone for the week. He described Palm's vision of the "Mobile Internet", where people have access to on-line resources no matter where they are.

Slides and WebCasts of the many of the conference presentations are available at <http://www.palmsource.com>.

### NEWS

Palm is currently riding high, with an 83% (US) market share in the handheld computer market. This was reflected in the list of Palm OS licensees. There are large consumer electronic companies like Sony, computer manufacturers such as IBM, and Palm-only companies like Handspring and TRG. Symbol is making Palm OS-based bar code scanners for use in enterprise applications.

They have created partnerships with several cell phone makers: Kyocera, Nokia, Motorola, and Samsung. Palm is working with these companies to put Palm OS capabilities into the next-generation (also known as 3G) cell phones.

Palm announced at the show that they have more than 130,000 registered developers, up from about 20,000 last year.

### PALM OS 4.0

Each developer (after signing an NDA) received a prerelease version of Palm OS 4.0. There were several sessions devoted to features that will be introduced in Palm OS 4.0, and how to take advantage of them. Slides of many of these

presentations are available on the PalmSource website. Some of the new features are:

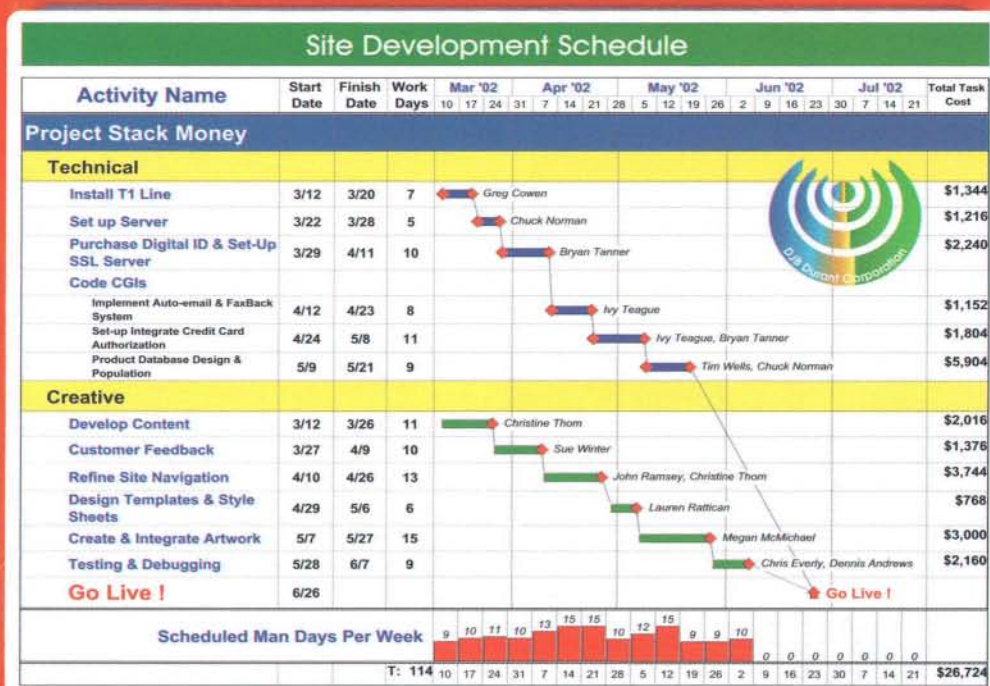
- **Secondary Storage**  
Palm OS 4 will support external file systems and secondary storage. This will allow users to keep data on devices such as smart cards or CompactFlash. Sony, a Palm OS licensee, was showing a Palm OS device with a slot for a MemoryStick.
- **Attention Manager**  
As Palm OS devices are used for more and more tasks, more and more applications need to get the user's attention. Under Mac OS, this is done using the Notification Manager. In Palm OS, applications use the Attention Manager. This allows the user to respond to single or multiple events, and allows the applications to prioritize notifications (simple vs. insistent, for example).
- **Exchange Manager**  
In Palm OS 3 and before, the Exchange Manager was tied to the IR port, since that was the only way that people could exchange data. In 4.0, exchange is transport-independent. The big driver for this, as far as I can see, is Bluetooth. However, this will also allow people using pager cards, SMS, etc. use the same mechanism as the built-in IR port.
- **Telephony**  
Since Palm is building relationships with several mobile phone vendors, you should not be surprised to learn that Palm OS 4.0 will contain a full "phone control" API. As well as the basic functionality, there will be also support for SMS (via the Exchange Manager).

### PALM OS 5.0

As well as 4.0, Palm gave attendees a glimpse of a future OS that they were calling Palm OS 5.0. (this is not an official name) David Fedor of Palm said that they expected that devices would ship with this OS in 2002. A few of the features previewed are:

Marshall has worked for Palomar Software, HP, Aladdin Systems and Adobe. Among other things, he has written PICT Detective, Aladdin's Resource Compression Toolkit, and way too many resource-processing tools. When he's not coding, he can be found mountain biking with his kids or checking out microbreweries. He can be reached at <[marshall@idio.com](mailto:marshall@idio.com)>

# MAGIC Etch A Schedule SCREEN



**FastTrack**  
Version 7 Schedule

THINK THIS IS THE  
ONLY WAY TO MAKE  
PROJECT SCHEDULING EASY?  
Think again.



So maybe scheduling a project isn't as simple as pushing a button or turning a knob. But then, neither is your development work. With FastTrack Schedule 7.0, you'll shave hours off the time it takes to organize, track, and manage all your development projects. And your eye-popping schedules are sure to turn heads and get results.



Whether the project on tap is coding the next killer app or putting a fresh coat of paint on a client's web site, FastTrack Schedule's three powerful views display your information the way you want—as a schedule, a calendar, or as a resource graph that tracks the people, equipment, and materials crucial to project success.

And with our new compatible Palm OS version, you can sync schedules between your desktop and handheld. So even when you're on the run, your schedules are right at your fingertips. FastTrack Schedule also includes FastSteps™ intra-application scripting, full support for AppleScript, and is available in a compatible Windows version. For a free demo version or to order, call us today at **800.450.1983** or visit **www.aecsoft.com**.

**Easily the best in Project Scheduling!**



© 1999 AEC Software, Inc. All rights reserved. Corporate names and trademarks stated herein are the property of their respective owners. Outside the United States, please call 703.450.1983 for information. The Etch A Sketch® product name and the configuration of the Etch A Sketch® product are trademarks of The AEC Software, Inc.

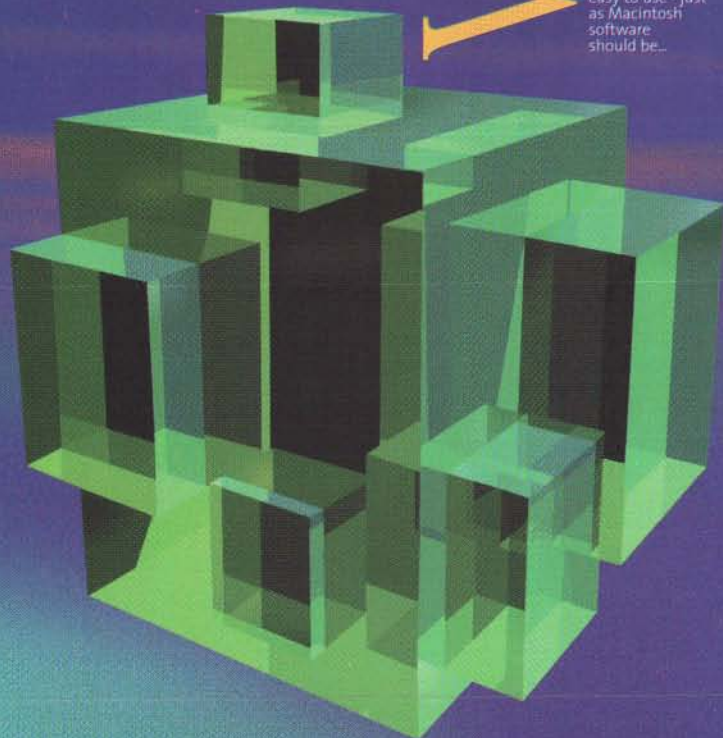


# The Version Control Tool for CodeWarrior Developers

# VOODOO Server



The successor of our award-winning VOODOO. More powerful and still easy to use - just as Macintosh software should be...



**www.unisoftwareplus.com**  
voodoo@unisoftwareplus.com

Available  
@ **DEPOT**®  
877-DEPOT-1X or 800-494-9797  
Fax: 800-644-9798  
http://www.depot.com  
orders@depot.com

- Support for ARM processors
- Better multimedia support
- Support for different screen sizes and resolutions
- A wide range of devices, both in price and capabilities.

The big change for 5.0 is the change to the ARM processor. This will be quite an undertaking for Palm, since all existing Palm OS applications are written for the Motorola 68000 processor. In some ways, this will be even more difficult than Apple's switch from the 680x0 to the PowerPC because the ARM is "little-endian", like the 80x86, rather than "big-endian", like the 68K and PowerPC. Palm will be supplying a 68K emulator in their new devices, so that existing applications can be run on new machines.

## DEVELOPER TOOLS

Metrowerks announced at the show (and shipped soon thereafter) CodeWarrior 7 for Palm OS. It includes an updated C/C++ compiler, all the IDE enhancements from CodeWarrior 6 for Mac OS and Windows, and several Palm OS-specific tools, including a debugger plugin that lets you symbolically debug your Palm OS applications either inside POSE or on an actual device. CodeWarrior for Palm OS runs on both the Mac and Windows.

Developer's Nation has formed a partnership with Palm to maintain and extend the Palm OS knowledge base, which is a great resource for Palm OS developers. This online Q&A database should be available by the time you read this at <http://www.DevNation.net>.

AppForge announced the release of Visual Basic for Palm OS, allowing developers who prefer to write their applications in Basic to create Palm OS applications. More information is available at <http://www.appforge.com>.

Extended Systems <<http://www.extendsys.com>> announced a set of developer tools geared towards the enterprise market, for integrating Palm OS devices into a medium to large company.

Last year at PalmSource '99, Palm and Bear River announced that they would be developing and releasing the "Palm Development Framework", a C++ class library for developing Palm OS applications. This year, Bear River announced that they had completed the Framework, which is available at <http://www.bearriver.com/developer/palm/>.

## SUMMARY

Last year, at PalmSource '99, I thought that this was an exciting time to be a Palm OS developer. If anything, this year promises to be even more exciting. Even though Palm has an 80% market share, they aren't sitting on their laurels; they are working hard on improving their products.

Last year, Palm beat the drum about enterprise support. Most of the sessions at PalmSource '99 were about getting into and supporting the enterprise market. This year while Palm talked quite a bit about the enterprise market, there were "personal" sessions as well.

If you couldn't make it to PalmSource 2000, be sure to check out <http://www.PalmSource.com> where slides for most of the sessions are available.

See you at PalmSource '01 in October!

By Rich Allen

Edited by Cal Simone

# Chart generation with AppleScript & Excel

A friend of mine likes to say “everything is easy once you know how”, Updating an Excel spreadsheet and generating an accompanying chart is easy using a little AppleScript, and Microsoft Excel, once you know how!

People like charts. Charts are an easy way of seeing a lot of data and understanding it with a quick glance. Using AppleScript to update an existing workbook, a simple VisualBasic macro for exporting a chart as a JPEG file and the iDo Script Scheduler included with OS 9 is all you need to create a regularly updated chart. This article will explain how to

- Use AppleScript to parse a data file for required values
- Transmit those values to Excel
- Tell Excel to export a chart as a JPEG file for use as part of a web page
- How to setup iDo so the chart is updated regularly

A note on conventions used in the AppleScript source listing; those words that have an underline are application keywords. These terms have a specific meaning to either the enclosing `tell` block or to AppleScript itself. See the preferences panel of your script editor for implementing this option.

First let's look at using AppleScript to open a file of raw data and parse out the values that we really want. For our raw data file, we will use a tab-delimited text file as shown in Listing 1.

## Listing 1: Sample raw data

| Date     | Hour | PhoneNum | Type | Members | Peg | Busy | Usage |
|----------|------|----------|------|---------|-----|------|-------|
| 12/20/00 | 09   | 5551234  | DNH  | 3       | 7   | 0    | 17    |
| 12/20/00 | 09   | 5552345  | DNH  | 4       | 9   | 0    | 34    |
| 12/20/00 | 09   | 5553456  | DNH  | 2       | 1   | 0    | 3     |
| 12/20/00 | 09   | 5554488  | DNH  | 4       | 5   | 0    | 6     |

For this project, we require the values under the headings labeled “Date”, “Hour”, “Peg” (number of telephone calls), and “Usage” (connect time of all calls) for telephone number 5554488. We will assume that this raw file is updated once each hour, although we will want to verify this, and that our chart will track peg values versus usage value count on an hourly basis. The Excel workbook will be pre-built with two worksheets; one containing the chart (chart) and another with the values that feed that chart (data). See **Figures 1** and **2** for examples of these worksheets.

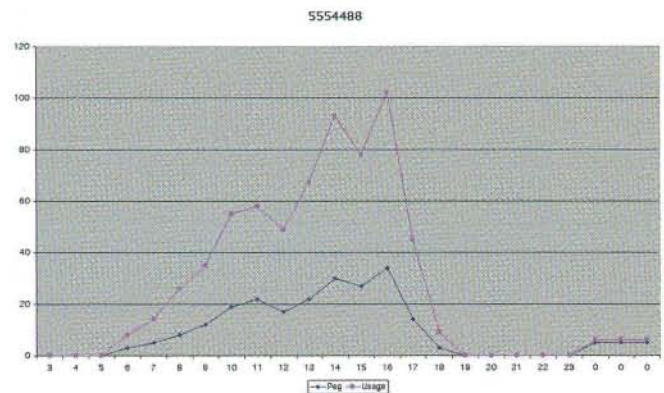


**Figure 1.** Sample chart output.

**Rich Allen** currently works for a local telephone company in Alaska in the traffic-engineering department. He has held a variety of telecommunications positions since 1984 and has designed a number of Macintosh based data systems. His email address is g3pb@alaska.net.

# RACKMOUNT YOUR MACINTOSH



## PowerRack

Rackmount chassis for PowerMac G3 and G4



## G-Rack

Rackmount solution for PowerMac G3 and G4 tower cases



## iRack

1RU Rackmount chassis for iMac

Rackmount solutions and  
accessories designed  
for the Macintosh

800-832-6326

www.marathoncomputer.com

MacTech@marathoncomputer.com

# MARATHON COMPUTER

|   | A        | B    | C   | D     |
|---|----------|------|-----|-------|
| 1 | Date     | Hour | Peg | Usage |
| 2 | 01/01/01 | 8    | 0   | 0     |
| 3 | 01/01/01 | 9    | 3   | 10    |
| 4 | 01/01/01 | 10   | 7   | 22    |
| 5 | 01/01/01 | 11   | 13  | 42    |
| 6 | 01/01/01 | 12   | 20  | 64    |

Figure 2. Property declaration.

OK, let's start writing a script! The script will use three properties, values that will be used throughout the script. These properties designate the path to the folder containing our files, the name of the raw data file, and the telephone number to find. See Listing 2.

### Listing 2: Sample of worksheet data

```
property folderPath : "MacHD:Desktop Folder:MacTech:"
property fileName : "rawdata"
property phoneNumber : "5554488"
```

It is always a good idea to have a `try` block around scripting statements to catch any error that may arise. The main portion of the script, Listing 3, is completely enclosed in a single `try` block. Any error encountered will be handled by the `on error` section. For example, if the raw data file can not be found when AppleScript tries to open it, an error will occur, passing control to the `on error` routine.

The script starts by setting `fileRef` to an empty string. By doing this, if an error does occur, the error handler will be able to determine if the data file is open, if so, closing it. If the error routine is invoked and `fileRef` is the empty string, then the error was generated while attempting to open the file thus there will be no need to execute the `close access` statement.

Next the script will attempt to open the raw data file, concatenating the `folderPath` and `fileName` properties into a single string as input for the `open for access` statement and assigning the resulting file reference number to the variable `fileRef`. From this point on, we can access data from the file by referring to the `fileRef` variable. We know that the data file is tab-delimited (a tab character separates each value in the file), so next, the script is set to use the tab character as its text item delimiter. (Normally AppleScript's text item delimiter is set to an empty character ("")). Since we already know that each value of interest per line is separated by the tab character, by setting AppleScript's text item delimiters to a tab we can easily refer to any value by its position.)

The sample data file contains eight items per line; in this case the date, the hour, phone number, phone number type, number of members associated with the phone number, a peg count, number of busies and the usage. The variable `num` will be set to the empty string and will later be set to the telephone number from each line that is read from the data file and then compared to the property `phoneNumber` to determine a positive match.

# Quickly Build and Deploy Powerful Data-Driven Web Applications



## Lasso Studio and Lasso Web Data Engine™ Lead The Way.

Building custom and shrink-wrapped database-driven Web applications requires a whole new way of doing things. Having pioneered the Web Data Engine™ over three years ago, Blue World and the Lasso Web Data Engine consistently lead the way providing a feature set Web developers describe as "incredible." Build online stores, discussion forums, resource management systems and other demanding database-driven Web applications with unrivaled performance, ease, security, extensibility, control and flexibility. Develop using multiple languages—including LDML, CDML, Server-Side JavaScript, Java and XML—and deploy across multiple platforms. Your Lasso code works identically regardless to which database you're connected. Lasso solutions for FileMaker® Pro databases easily scale to big iron ODBC-compliant databases like Oracle, Informix, Sybase and more with little or no change. What's more, the Lasso Java Application Programming Interface (LJAPI) provides developers an easy-to-use Java-based API for unprecedented extensibility.

Find out why hundreds of thousands of websites rely on award-winning Lasso technology for their business critical Web data. Download a 30-day evaluation copy at [www.blueworld.com/download/](http://www.blueworld.com/download/) or order securely online today at the Blue World Store at [store.blueworld.com](http://store.blueworld.com).

**Lasso Product Line – The leading Web tools for Macintosh and beyond.**

bring business to the internet.™

**blueworld**

At this point the script is ready to start reading each line of the data file until it finds a match of the phone number read and the desired number. Reading a line of the data file is done with the standard `read` statement; reading from the current file position until the next encountered return character. If the two telephone numbers do not match, the next line is read and phone numbers compared. If the desired phone number is not found before the end of the file (eof) is reached then an error will be created by AppleScript and handled by the `on error` portion of the run handler, otherwise the `close` statement will be executed to close the data file.

### Listing 3: The run handler

```
try
  set oldDelimiter to AppleScript's text item delimiters
  set AppleScript's text item delimiters to tab
  set fileRef to ""
  set fileRef to open for access (folderPath & fileName)
  set num to ""
  repeat until num is phoneNumber
    set dLine to read fileRef before return
    set num to text item 3 of dLine
    if (phoneNumber is num) then
      UpdateWorkbook(dLine)
    end if
  end repeat
  close access fileRef
  set AppleScript's text item delimiters to oldDelimiter
on error errMsg number errNum
  if fileRef is not "" then
    close access fileRef
  end if
  set AppleScript's text item delimiters to oldDelimiter
end try
```

Once a match has been found, the `UpdateWorkbook` handler (Listing 4) is called. (A handler is basically equivalent to a subroutine.) The script starts with a `tell` statement that will direct any statements following towards Excel. For any statement within the `tell` block that are not directly understood by Excel, those statements will be handled by AppleScript itself (e.g. `if`).

The first statement to Excel will open the required workbook by combining two of the properties set initially and adding the ".xls" suffix to match the exact file name and path of the required workbook. Since the workbook contains more than one worksheet (chart and data), the script selects the data worksheet to assure the values from the raw data file will be inserted into the proper cells.

For this workbook and chart we are tracking 24 hours of data. The data worksheet contains a header in the first row with the following 24 rows containing the last 24 hours worth of data. Refer back to Figure 2 for an example.

We will want to verify that the raw data we have read is new before we update the worksheet. By comparing the hour from the line of raw data with the last hour on the data worksheet (cell B25), we can be sure that the raw data has been changed.

Since we want our chart to show peg versa usage information on an hourly basis, we will want to drop the oldest hour's data from our chart and add the newest data. By copying the last 23 hours data values (rows 3 through 25) and then pasting those values into the first 23 hours

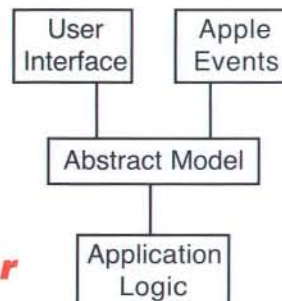
## Fight Boredom

Let's face it: Much of programming is boring and repetitive. Well, that's where the right tool can save days, weeks, or even months of your valuable time.

## AppMaker

**Your Assistant Programmer**

AppMaker makes it faster and easier to make an application. It's like having your own assistant programmer. You point and click to tell AppMaker the results you want, then it generates "human, professional quality code" to implement your design.



### Model-View-Controller

AppMaker's generated code uses the MVC paradigm. It separates the user interface from application logic, making code easier to write. You deal only with abstract data; AppMaker takes care of the user interface.

### Scriptable Applications

AppMaker generates the 'aete' resource and generates code to access your data (Properties and Elements in the Apple Event Object Model) and to handle Events.

Just \$199 from [www.devdepot.com](http://www.devdepot.com)

**B•O•W•E•R•S**  
**Development**

P.O. Box 929, Grantham, NH 03753 • (603) 863-0945 • FAX 863-3857  
[bowersdev@aol.com](mailto:bowersdev@aol.com) • <http://members.aol.com/bowersdev>

data (rows 2 through 24), the script has effectively dropped the oldest hour. (Note that the `copyobject` keyword, used here, comes from Excel's Custom Suite — unfortunately Excel does not have the standard `copy` verb as many other applications do.) Rows 24 and 25 are now the same. To add the newest data, the script will simply set the new values in row 25, overwriting the extraneous values.

Now that all the values have been updated, it's time to export a copy of the chart for use on the web page. Excel has an export function using VisualBasic that can accomplish this (the macro is contained within the workbook). We will not delve into Excel VisualBasic macros in this article but will show the macro we're using here in Listing 5. AppleScript uses the `evaluate` statement to communicate with Excel to run the macro. Be sure to note the syntax here, `evaluate` requires the file name of the workbook concatenated with the name of the macro with the exclamation point character separating them.

The last job for the `UpdateWorkbook` handler is to save the changes made to the workbook and close the workbook file.

#### Listing 4: The `UpdateWorkbook` handler

```
on UpdateWorkbook(dataLine)
    tell application "Microsoft Excel"
        open (folderPath & phoneNumber & ".xls")
        select sheet "data"
        if (text item 2 of dataLine ≠ value of cell "$B$25") then
            select range ("A$3:$D$25")
            CopyObject selection
            select range ("A$2:$D$24")
            paste
            set value of cell "A$25" to text item 1 of dataLine
            set value of cell "B$25" to text item 2 of dataLine
            set value of cell "C$25" to text item 6 of dataLine
            set value of cell "D$25" to text item 8 of dataLine
            evaluate (phoneNumber & ".xls!ExportChart()")
            close ActiveWorkbook saving yes
        end if
    end tell
end UpdateWorkbook
```

#### Listing 5: VisualBasic macro to export chart as JPG

```
Sub ExportChart()
    Charts("Chart").Export _
        FileName:="pb0\Desktop"
    Folder:MacTech:5554488.jpg"
End Sub
```

Each time the script is run it will update the workbook and export a new chart for use on a web page. This script does not provide a method of performing this task on a regular basis so how do we get it to run on a schedule? One solution is to use the iDo Script Scheduler that is included with the Mac OS. If it is not already in your control panels folder, you will find it on the Mac OS 9 CD in the CD Extras folder in the AppleScript Extras folder.

Open the iDo Script Scheduler control panel and click on the New button. Give this event a name, set the trigger to "repeating", and select the start time and date. All that is left is to click the Choose button, select the script (which we've saved as a compiled script), and save the newly created event. Your chart will now be updated every hour at the specified time. See **Figure 3** for an example of the iDo event.

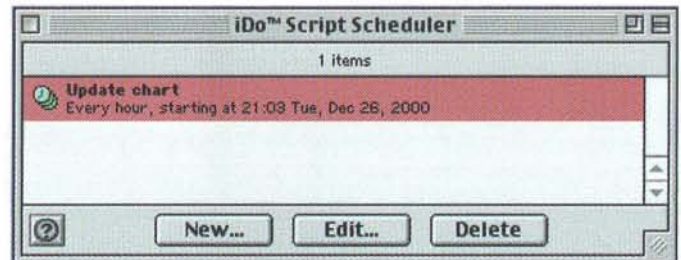
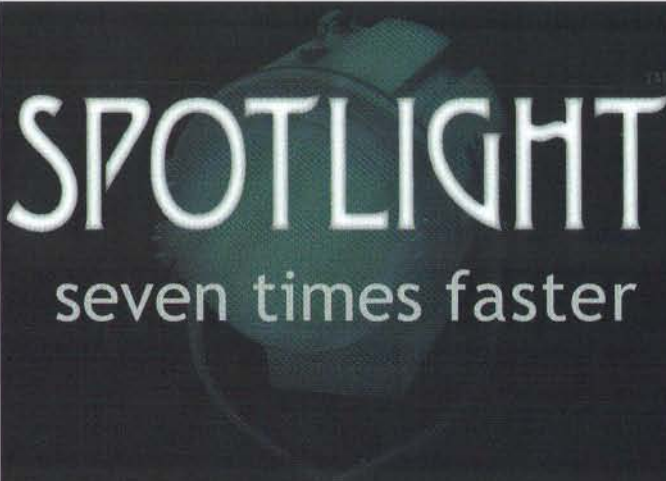


**Figure 3.** iDo event.

Although this is a short script with only limited error checking, it does demonstrate the ease of integrating AppleScript with Microsoft Excel to create charts suitable for use on any web page!

My friend also has another thing he likes to say, "Now that I know how, it's easy!" 



# SPOTLIGHT

## seven times faster

free demo  
[www.onyx-tech.com](http://www.onyx-tech.com)

Find memory errors automatically in source  
Code Fragment Support  
Leak Detection  
Toolbox Parameter Checking




by Bob Boonstra, Westford, MA

## DRAGSORT

I'll confess. I've done it. More than once, actually. It's probably not legal, strictly speaking. But how bad can it be. I mean, lots of people must do it. It's not something truly evil like, well, like Napster.

What am I talking about? Downloading music. From the internet. Music that other people posted, music that I haven't purchased. Yet, that is. Music posted to the UseNet alt.binaries.sounds.\* newsgroups.

What does this have to do with the Programmer's Challenge? Downloading Usenet binaries is a (perhaps contrived) motivation for this month's problem, that of sorting a list of items in a particular way. Large binaries are posted in a sequence of parts, and newsreaders thread the parts based on sequence information included by convention in the subject line. But sometimes the subject line contains information that confuses the sequencing, meaning that the parts need to be sorted manually before the binary information can be extracted. With the newsreader I use, the parts are sorted by dragging items from one position in the list to another position, until the parts appear in the correct order.

Your Challenge is to find an efficient way to perform this sort. Efficient in this case means minimizing the amount of tiresome clicking and dragging needed to put the parts in the correct order. Specifically, you want to minimize the cumulative number of positions that you need to move to select the part to be dragged, and the number of positions you need to drag the parts. An example in a later paragraph will make this a little clearer.

Oh, and before you turn me in to the authorities, downloading music from the internet allows one to sample the music before deciding whether to buy it. We all, of course, buy what we keep.

The prototype for the code you should write is:

```
typedef struct Move { /* describes moves used to sort itemsToSort */
    long selectPosition; /* select this item in the array, origin 0 */
    long dragToPosition; /* drag selected item to this position in the array, origin 0 */
} Move;

long /* numMoves */ DragSort(
    long itemsToSort[], /* array of items to be sorted */
    long numItemsToSort, /* number of itemsToSort */
    long startPosition, /* item initially selected */
    Move sortMoves[] /* store Moves that sort the array here */
);
```

Your **DragSort** routine will be called with a list of **itemsToSort**, a count of the number of items in the list (**numItemsToSort**), and the element of the list initially selected (**selectPosition**). **DragSort** should calculate a sequence of **Moves** that reorder the list into ascending order by moving the item located at one position in the list (**selectPosition**) to another position in the list (**dragToPosition**). **Moves** are returned in the **sortMoves** array, and the number of **Moves** needed to sort the array should be returned by **DragSort**.

Sounds simple and boring, right? The catch is in how your sort solution is scored. You are penalized one point for each position move your solution makes while performing the sort – one point for each position between your current location and the **selectPosition** of the next **Move**, and one point for each position between that **selectPosition** and the **dragToPosition**. When a **Move** is completed, your current position is the **dragToPosition**. The penalty for the next **Move** is the number of positions between the previous **dragToPosition** and the current **selectPosition**, plus the distance from the current **selectPosition** and the current **dragToPosition**, etc.

Imagine, for example, that you are asked to sort the following list, with a **startPosition** of 0:

6 1 2 3 5 4

You might employ a bubble sort. Your first **Move** is to select the 2nd item in the list and move it to the 1st position, at a cost of two penalty points (moving from [0] to [1], and dragging from [1] to [0]). The list now looks like this:

1 6 2 3 5 4

Next you might move the 3rd item to the 2nd position, at a cost of 3 penalty points (moving from [0] to [2], and dragging from [2] to [1]), leaving the list like this:

1 2 6 3 5 4

Then you might move the 4th item to the 3rd position, at a cost of 3 more penalty points. Then the 6th item to the 4th position (cost 5 points), and the 6th item to the 5th position (cost 3 points). The list would then be correctly sorted, at a cost, if I have counted correctly, of 16 points.

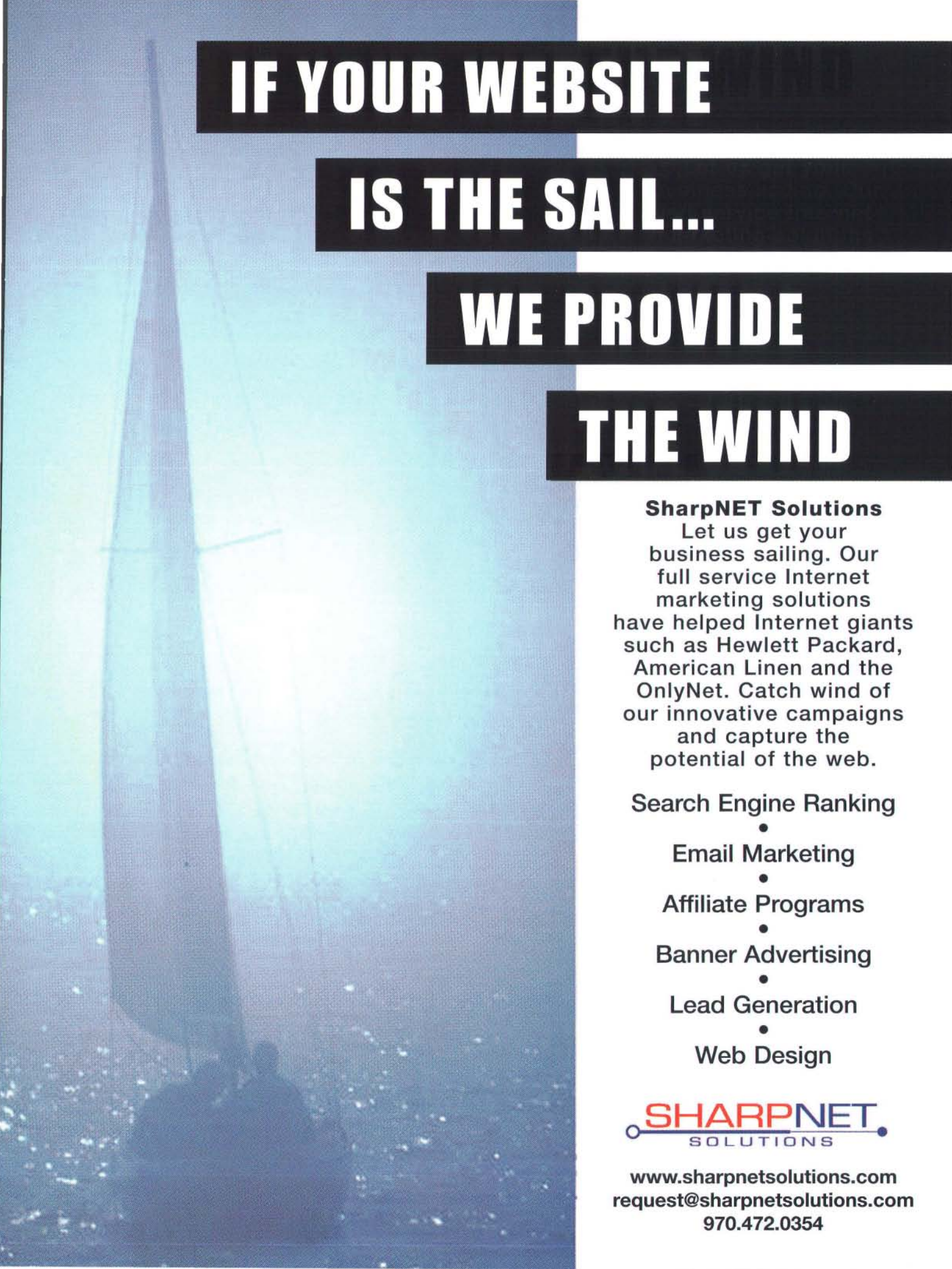
And you would lose the Challenge.

A more successful contestant would sort the list at a cost of 7 points as follows:

```
6 1 2 3 5 4
1 2 3 5 4 6 (cost 5 points)
1 2 3 4 5 6 (cost 2 points)
```

In addition to the points incurred for dragging list items around, a penalty of 1 point will be assessed for each millisecond of execution time. The winner will be the entry that correctly sorts a sequence of lists while accumulating the fewest penalty points. The Challenge prize will be divided between the overall winner and the best scoring entry from a contestant that has not won the Challenge recently.

This will be a native PowerPC Challenge, using the CodeWarrior Pro 6 environment. Solutions may be coded in C, C++, or Pascal. You may provide a solution in Java instead, provided you also provide a test driver equivalent to the C code provided on the web for this problem.



**IF YOUR WEBSITE**

**IS THE SAIL...**

**WE PROVIDE**

**THE WIND**

**SharpNET Solutions**

Let us get your business sailing. Our full service Internet marketing solutions have helped Internet giants such as Hewlett Packard, American Linen and the OnlyNet. Catch wind of our innovative campaigns and capture the potential of the web.

Search Engine Ranking



Email Marketing



Affiliate Programs



Banner Advertising



Lead Generation



Web Design



[www.sharpnetsolutions.com](http://www.sharpnetsolutions.com)  
[request@sharpnetsolutions.com](mailto:request@sharpnetsolutions.com)  
970.472.0354

### THREE MONTHS AGO WINNER

Congratulations to **Tom Saxton** for winning the December Crutches Challenge. Motivated by a broken foot injury to a member of the family, this variation on the traveling salesperson problem required contestants to find optimal way of moving a set of objects from one location to another, operating with connectivity constraints among the locations, and limited by a maximum carrying capacity. The object was to minimize a score that was a combination of the length of the path traversed and the amount of execution time required to complete the solution. Tom's entry was not the fastest, but it did compute a more optimal path than the second-place entry by Ernst Munter.

Tom first calculates the distance from each node to each other node in the `_MapConnections` function. He then calculates a baseline solution in the `_CatCrutchesCore` routine, a solution that picks up the closest object when nothing is being carried, delivers the closest object when something is being carried, and tries to move objects closer to their eventual destination when there is spare carrying capacity. Tom then successively refines the baseline solution until a time allocation is exceeded. Refinement uses a variant of the "Algorithm of the Gods", published in Scientific American in March, 1997 (see <http://www.scientificamerican.com/0397issue/0397amsci.html>). Basically, the algorithm makes small perturbations in the baseline solution to see if the perturbed solution results in a lower cost.

Ernst's entry was approximately three times faster than Tom's, but it resulted in less optimal solutions. Actually, Ernst's solutions were slightly shorter in approximately half of the test cases, but Tom's were shorter by 1/3 to 1/2 in the larger test cases, giving him the win.

I used 12 test cases for evaluation, where the number of nodes ranged from 10 to 200, the number of connections was as high as 2200, and the number of tasks to be accomplished ranged from 20 for the small graphs to 500 for the larger graphs.

The table below lists, for each of the solutions submitted, the total execution time, the cumulative distance traveled to accomplish all of the delivery tasks, and the number of points earned. As points were incurred based on distance traveled and elapsed time, a smaller number of points was better. As usual, the number in parentheses after the entrant's name is the total number of Challenge points earned in all Challenges prior to this one. The solution marked with an asterisk did not complete the larger problems in a reasonable amount of time.

| Name               | Total<br>Time (msec) | Distance<br>Moved | Total<br>Points | Code<br>Size | Data<br>Size | Lang |
|--------------------|----------------------|-------------------|-----------------|--------------|--------------|------|
| Tom Saxton (165)   | 12773.5              | 456815            | 512018          | 8004         | 586          | C++  |
| Ernst Munter (701) | 4408.6               | 557477            | 596511          | 4852         | 324          | C++  |
| J. S.              | *                    | *                 | *               | 5948         | 344          | C++  |

### TOP CONTESTANTS ...

Listed here are the Top Contestants for the Programmer's Challenge, including everyone who has accumulated 20 or more points during the past two years. The numbers below include points awarded over the 24 most recent contests, including points earned by this month's entrants.

| Rank | Name              | Points | Rank | Name             | Points |
|------|-------------------|--------|------|------------------|--------|
| 1.   | Munter, Ernst     | 281    | 5.   | Boring, Randy    | 52     |
| 2.   | Saxton, Tom       | 96     | 6.   | Shearer, Rob     | 48     |
| 3.   | Maurer, Sebastian | 68     | 7.   | Taylor, Jonathan | 36     |
| 4.   | Rieken, Willeke   | 65     | 8.   | Wihlborg, Claes  | 29     |

### ... AND THE TOP CONTESTANTS LOOKING FOR A RECENT WIN

In order to give some recognition to other participants in the Challenge, we also list the high scores for contestants who have accumulated points without taking first place in a Challenge during the past two years. Listed here are all of those contestants who have accumulated 6 or more points during the past two years.

| Rank | Name              | Points | Rank | Name             | Points |
|------|-------------------|--------|------|------------------|--------|
| 9.   | Downs, Andrew     | 12     | 17.  | Strout, Joe      | 10     |
| 10.  | Jones, Dennis     | 12     | 18.  | Hala, Ladislav   | 7      |
| 11.  | Day, Mark         | 10     | 19.  | Miller, Mike     | 7      |
| 12.  | Duga, Brady       | 10     | 20.  | Nicollé, Ludovic | 7      |
| 13.  | Fazekas, Miklos   | 10     | 21.  | Schotsman, Jan   | 7      |
| 14.  | Flowers, Sue      | 10     | 22.  | Widyatama, Yudhi | 7      |
| 15.  | Sadetsky, Gregory | 10     | 23.  | Heithcock, JG    | 6      |
| 16.  | Selengut, Jared   | 10     |      |                  |        |

There are three ways to earn points: (1) scoring in the top 5 of any Challenge, (2) being the first person to find a bug in a published winning solution or, (3) being the first person to suggest a Challenge that I use. The points you can win are:

|                      |           |
|----------------------|-----------|
| 1st place            | 20 points |
| 2nd place            | 10 points |
| 3rd place            | 7 points  |
| 4th place            | 4 points  |
| 5th place            | 2 points  |
| finding bug          | 2 points  |
| suggesting Challenge | 2 points  |

Here is Tom's winning Crutches solution:

### Crutches.cpp

Copyright © 2000

Tom Saxton

```
//
// Crutches Challenge Solution, December 2000.
// (c) 2000, Tom Saxton
//
#include "Crutches.h"

#include <Events.h>

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

// the basics
enum { fFalse = 0, fTrue = 1 };

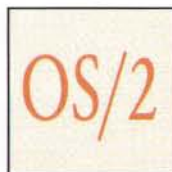
// disable asserts
#define Assert(f)

// how long should we spend trying to find a better answer?
#define dtickSecond 60
#define dtickMax (1*dtickSecond)
```

The *True* BASIC<sup>®</sup> story  
is a *powerful* ~~simple~~ story.

Write your code once.

Run it (without change)



here, here, here, here, here, here & here!

Demos and information from our web site:

<http://www.truebasic.com>

**TRUE BASIC INC** • Founded by the *Inventors* of BASIC

PO BOX 5428 • WEST LEBANON NH 03784-5428 • 800 436-2111 • 802 296-2711

```
// random
static unsigned short URand(unsigned short uMac);

// heap sort
typedef unsigned long INDEX;
typedef int (*PFNCMP)(INDEX i1, INDEX i2, void *pv);
extern void HeapSort(void *prgfoo, INDEX iMac, int cbFoo,
PFNCMP pfncmp);

// maximum distance value
#define distMax 0x7FFFFFFF

// structures for mapping problem to node indices
typedef struct TSK TSK;
struct TSK
{
    long itaskObject;
    long inodeStart;
    long inodeCur;
    long inodeDst;
    long wgt;
    int fWanted;
    long rankDrop;
    TSK *ptskNext;
};

typedef struct ENODE ENODE;
struct ENODE
{
    long inode;
    Node node;
    long cobj;
    long rankGrab;
};

// connection map stuff
typedef struct CONN CONN;
struct CONN
{
```

```
// base information for this connection
long inode1;
long inode2;
long dist;
};

typedef struct CMAP CMAP;
struct CMAP
{
    long distDirect;
    long distPath;
};

typedef struct NODC NODC;
struct NODC // node connection, used when looking for paths between nodes
{
    long distFromStart;
    int fCheckNeighbors;
};

static CONN *s_pacomm;
static long s_cconn;
static CMAP *s_pacmap;
static NODC *s_panodc;

// translation globals
static const Node *s_panode;
static ENODE *s_pdnnode;
static long s_cnode;
static TSK *s_ptask;
static TSK *s_ptskFirst;
static long s_ctsk;
static long s_wgtCarryMost;

// core algorithm stuff
static long _CactCrutchesCore(long inodeStart, Action paact[],
    long cactMax, long distStop, long *painodePath,
    long *paitskOrder, long *paitskOrderUsed, long *pdist);
static long _InodeGetTarget(long inodeCur, long cobjCarry,
    int fUseWantedFlag, long *paitskOrder, long
    *pcnodeOrderGrab,
    long *petskOrderDrop);
static void _RandomizeOrder(long iTry, long cTry, long ctsk,
    long cnode, const long paitskOrderOld[],
    long paitskOrderNew[]);
```

```
// translation helpers
static long _InodeFromNode(Node node);
static int _CmpEnodeNode(INDEX i1, INDEX i2, void *pv);
static int _CmpEnodeInode(INDEX i1, INDEX i2, void *pv);

// action functions
static int _FTakeStep(long inodeDst, long cactMax,
    Action paact[], long *pcact, long *pinodeCur, long *pdist);
static int _FFollowPath(long cactMax, Action paact[],
    long *pcact, long *pinodeCur, long *pwgtCarry,
    long *pcobjCarry, long cnodePath, const long painodePath[]);
static int _FGrabObject(long cactMax, Action paact[],
    long *pcact, long *pwgtCarry, long *pcobjCarry, long inode,
    long itsk);
static int _FDropObject(long cactMax, Action paact[],
    long *pcact, long *pwgtCarry, long *pcobjCarry, long inode,
    long itsk);

// connection map functions
static void _MapConnections(long cconn,
    const Connection paconnection[]);
static void _FreeConnectionMap(void);
static int _FConnectedNodes(long inode1, long inode2,
    long *pdist);
static long _CnodeFindMinPath(long inodeStart,
    long inodeEnd, long painodePath[]);
static void _ComputeNodeDistances(long inode);
static long _InodeNearestTask(long inodeCur);
static inline long _DistNodeToNode(long inode1, long inode2)
{
    return s_pacmap[inode2*s_cnode + inode1].distPath;
}

static void _ResetState();
static TSK *_PtskRemoveTsk(TSK *ptsk);
```

Crutches

## DATA RECOVERY: 800-440-1904

*"Their incredibly kind staff focus on getting the job done as well as their customer's feelings. All experiences should be so pleasant."*

—Neil Tickin, Publisher  
MacTech Magazine

### 7 Good Reasons to Choose DriveSavers



1. Fastest, most successful data recovery service available.
2. Retrieve recovered data instantly with DATAEXPRESS™ over high speed secured Internet lines.
3. Authorized to perform Data Recovery on all Apple computers and hard drives.
4. 24-hour, onsite, and weekend services.
5. Advanced, proprietary recovery techniques.
6. Featured by CNN, BBC, Forbes. Also in Mac World, Mac Addict, Popular Mechanics, and many others.
7. Federal and State Contracts (GSA, CMAS).



**"We Can Save It!"**

INTL: 415-382-2000  
www.drivesavers.com

**DriveSavers – Your Data Recovery Solution!**

©2000 DRIVESAVERS, INC. 400 BEL MARIN KEYS BLVD., NOVATO, CA 94949, INTL: 415-382-2000, FAX: 415-883-0780

**Installer VISE 7.1**  
**for Mac OS X**



```

long /* actions in solution */ Crutches (
    const Node panode[], /* Nodes defining the problem */
    long cnode, /* number of Nodes */
    const Connection paconnection[], /* Connections between nodes */
    long cconn, /* number of Connections */
    const Task patask[], /* objects to be moved */
    long ctask, /* objects to be moved, numbered as index into objectsToMove */
    Node nodeStart, /* start from this node */
    Weight wgtCarryMost, /* maximum weight that you can carry */
    Action paact[], /* return your solution here */
    long cactMax, /* size of solutionPath array */
)
{
    long cactBest = 0;
    long *paitskOrderTry = NULL;
    long *paitskOrderBest = NULL;
    long *paitskOrderUsed = NULL;
    long *painodePathBuffer = NULL;

    // start the clock running...
    long tickStart = TickCount();

    // allocate the memory buffers we need (_MapConnections also allocates memory)
    try
    {
        s_pdnnode = new ENODE[cnode];
        s_patsk = new TSK[s_ctsk = ctask];
        paitskOrderTry = new long[ctask+cnode];
        paitskOrderBest = new long[ctask+cnode];
        paitskOrderUsed = new long[ctask+cnode];
        painodePathBuffer = new long[s_cnode];
    } catch (...) {
        printf("Bob, can I have some more memory?
            I need %ld bytes, plus some more for the
            connections map.\n",
            sizeof(ENODE)*cnode
            +sizeof(TSK)*ctask
            +sizeof(long)*(ctask+cnode)*3
            +sizeof(long)*s_cnode
            );
        throw;
    }
}

```

## Engineer for Hire

**Henry Carstens**  
Contract Software Engineer

Cross-platform C, C++  
CodeWarrior, PowerPlant  
Classic, Carbon, OSX

13 Years Experience  
20 Commercial Software Apps Shipped  
Fast, Dependable, Self-Motivating

On Site or Off

**Henry Carstens**  
carstens@verticalsolutions.com  
503-524-6831

Because how soon you ship *does* matter.

```

if (s_pdnnode == NULL || s_patsk == NULL ||
    paitskOrderUsed == NULL || paitskOrderTry == NULL ||
    paitskOrderBest == NULL || painodePathBuffer == NULL)
    goto LCleanup;

// initialize the node entry array
s_cnode = cnode;
s_panode = panode;
for (long inode = 0; inode < cnode; ++inode)
{
    ENODE *penode = &s_pdnnode[inode];
    penode->inode = inode;
    penode->node = panode[inode];
    penode->cobj = 0;
}

// sort node entries by node value
HeapSort(s_pdnnode, cnode, sizeof(s_pdnnode[0]),
    _CmpENodeNode);

// fill in TSK records, with inode values
for (long itask = 0; itask < ctask; ++itask)
{
    TSK *ptsk = &s_patsk[itask];
    const Task *ptask = &patask[itask];

    ptask->itaskObject = itask;
    ptask->inodeStart = _InodeFromNode(ptask->fromNode);
    ptask->inodeDst = _InodeFromNode(ptask->toNode);
    ptask->wgt = ptask->weight;
}

_MapConnections(cconn, paconnection);

long inodeStart = _InodeFromNode(nodeStart);

// sort node entries by node index
HeapSort(s_pdnnode, cnode, sizeof(s_pdnnode[0]),
    _CmpENodeInode);

// get the baseline solution
_ResetState();
s_wgtCarryMost = wgtCarryMost;
long distBest;
cactBest = _CactCrutchesCore(inodeStart, paact, cactMax,
    distMax, painodePathBuffer, NULL, paitskOrderUsed,
    &distBest);

// if we have a non-trivial number of tasks, try to improve our solution using partial
// ordering info from the baseline
// based loosely on "Algorithm of the Gods", Scientific American, March 1997
if (ctask > 2)
{
    long distToAccept = distMax;
    long cIter = 0;
    long cIterMax = 10 * cconn * ctask;
    long dtickNext = 0;

    BlockMove(paitskOrderUsed, paitskOrderBest,
        sizeof(paitskOrderBest[0])*(ctask+cnode));

    for (long dtick = TickCount();
        (dtick - tickStart) < dtickMax && cIter < cIterMax;
        ++cIter, dtick = TickCount())
    {
        _ResetState();

        _RandomizeOrder(dtick, dtickMax, ctask, cnode,
            paitskOrderBest, paitskOrderTry);

        long distTry;
        long cact = _CactCrutchesCore(inodeStart, paact+cactBest,
            cactMax-cactBest, distToAccept, painodePathBuffer,
            paitskOrderTry, NULL, &distTry);
        if (distTry < distToAccept)
        {
            distToAccept = distTry;
            BlockMove(paitskOrderTry, paitskOrderBest,
                sizeof(paitskOrderBest[0])*(ctask+cnode));

            if (distTry < distBest)
            {
                distBest = distTry;
                BlockMove(paact+cactBest, paact, cact*sizeof(paact[0]));
            }
        }
    }
}

```



```

long distBest = distMax;
for (ptskT = s_ptskFirst; ptskT != NULL;
    ptskT = ptskT->ptskNext)
{
    if (ptskT->fWanted ||
        (ptskT->inodeCur != inodeCur && ptskT->inodeCur != -1)
        || wgtWanted + ptskT->wgt > s_wgtCarryMost)
        continue;

    long distLook;
    if (paitskOrder != NULL)
        distLook = ptskT->rankDrop;
    else
        distLook = _DistNodeToNode(inodeCur, ptskT->inodeDst);
    if (distLook < distBest)
    {
        distBest = distLook;
        ptskWant = ptskT;
    }
}

// bail if we didn't find anything more to pick up
if (ptskWant == NULL)
    break;

// mark the new item as wanted
ptskWant->fWanted = fTrue;
wgtWanted += ptskWant->wgt;
cobjCarryWanted += 1;
}

// did anything change?
for (ptskT = s_ptskFirst; ptskT != NULL;
    ptskT = ptskT->ptskNext)
if ((ptskT->inodeCur == -1) != (ptskT->fWanted != fFalse))
    break;
if (ptskT != NULL)
{
    inodeTarget = _InodeGetTarget(inodeCur, cobjCarryWanted,
        fTrue/"fUseWantedFlag"/, paitskOrder, &cnodeOrderGrab,
        &ctskOrderDrop);
    if (inodePath != inodeTarget)

```

```

{
    // make sure we have a path to the current target
    inodePath = inodeTarget;
    if ((cnodePath = _CnodeFindMinPath(inodeCur, inodePath,
        painodePath)) == 0)
        goto LGiveUp;
    iinodePath = 0;
}
inodeNext = painodePath[iinodePath];

// clear items that aren't going to be helped by the path to the new target
for (ptskT = s_ptskFirst; ptskT != NULL;
    ptskT = ptskT->ptskNext)
{
    if (!ptskT->fWanted)
        continue;

    _ComputeNodeDistances(ptskT->inodeDst);
    if (_DistNodeToNode(ptskT->inodeDst, inodeNext) >
        _DistNodeToNode(ptskT->inodeDst, inodeCur))
    {
        ptskT->fWanted = fFalse;
        wgtWanted -= ptskT->wgt;
        cobjCarryWanted -= 1;
    }
}

// drop items we're carrying that we no longer want
for (ptskT = s_ptskFirst; ptskT != NULL;
    ptskT = ptskT->ptskNext)
if (ptskT->inodeCur == -1 && !ptskT->fWanted)
if (!_FDropObject(cactMax, paact, &cact, &wgtCarry,
    &cobjCarry, inodeCur, ptskT - s_patsk))
    goto LCleanup;

// grab wanted items we're not carrying
for (ptskT = s_ptskFirst; ptskT != NULL;
    ptskT = ptskT->ptskNext)
if (ptskT->inodeCur == inodeCur && ptskT->fWanted)
if (!_FGrabObject(cactMax, paact, &cact, &wgtCarry,
    &cobjCarry, inodeCur, ptskT - s_patsk))
    goto LCleanup;
}

// pick the current target node
if (inodeTarget == -1)
    inodeTarget = _InodeGetTarget(inodeCur, cobjCarry,
        fFalse/"fUseWantedFlag"/, paitskOrder, &cnodeOrderGrab,
        &ctskOrderDrop);

// make sure we have a path to the current target
if (inodePath != inodeTarget)
{
    inodePath = inodeTarget;
    if ((cnodePath = _CnodeFindMinPath(inodeCur, inodePath,
        painodePath)) == 0)
        goto LGiveUp;
    iinodePath = 0;
}
inodeNext = painodePath[iinodePath];
if (++iinodePath >= cnodePath)
    inodeTarget = inodePath = -1;

// drop any objects not helped by our pending move
if (cobjCarry > 0)
{
    for (ptskT = s_ptskFirst; ptskT != NULL;
        ptskT = ptskT->ptskNext)
    {
        if (ptskT->inodeCur != -1)
            continue;

        _ComputeNodeDistances(ptskT->inodeDst);
        if (_DistNodeToNode(ptskT->inodeDst, inodeNext) >
            _DistNodeToNode(ptskT->inodeDst, inodeCur))
        if (!_FDropObject(cactMax, paact, &cact, &wgtCarry,
            &cobjCarry, inodeCur, ptskT - s_patsk))
            goto LCleanup;
    }
}

// move

```

## StoneTable

You thought it was **just** a replacement  
for the List Manager ?

We lied, it is **much** more !

Tired of always adding just one more feature to your LDEF or  
table code ? What do you need in your table ?

Pictures and Icons and Checkboxes ?  
adjustable columns or rows ?  
Titles for columns or rows ?  
In-line editing of cell text ?  
More than 32K of data ?  
Color and styles ?  
Sorting ?  
More ??

How much longer does the list need to be to make it worth  
\$200 of your time ?

See just how long the list is for StoneTable.

Make StoneTable part of your toolbox today !

Only \$200.00

MasterCard & Visa accepted.

More Info & demo  
<http://www.teleport.com/~stack>

StoneTable Publishing  
Voice/FAX (503) 287-3424  
[stack@teleport.com](mailto:stack@teleport.com)

# Looking for more speed, scalability and reliability from your database? **c-tree Plus<sup>®</sup>** is the perfect fit!

What shape is your database project? A tiny embedded database that requires a small footprint but demands rigorous functionality? Or a huge multi-platform, multi-user, multi-headache project? Have you struggled to find a better solution for your data access requirements, only to feel like you're sacrificing one requirement over another?

FairCom has delivered uncompromising database technology to commercial developers for over twenty years, with the primary goal of keeping control in the hands of the developer. Our customers rely on the speed, flexibility, scalability and reliability of c-tree Plus. The high performance and low cost of ownership make c-tree Plus an excellent choice for all sizes of database development projects.

## Proven Database Technology

Besides the ISAM-level control and speed, perhaps the most important reason why small development houses and Fortune 1000 companies have chosen FairCom technology is the superior service offered by our sales and support staff. Our development staff understands the challenges you face, and we'll put our twenty years of experience to work to help you build better solutions.

## Single Solution for Diverse Implementation

- Vertical Markets
- Embedded Systems
- Web & ASP Markets
- E-Commerce
- Smart-Cards
- Web-Enabled Appliances

## Comprehensive Feature Set

- Royalty-free single user and multi-user support with full source code
- Client/server and custom server support
- Robust server side SDK - build your own application specific database server with full source available!
- Thread compliant + portable thread API
- Full featured transaction processing with savepoints, abort and full rollback
- Comprehensive security and encryption features
- Small footprint
- Fixed and variable length records and keys
- Store any data type - up to 18 million terabytes!
- Dynamic space reclamation
- ODBC and Crystal Reports<sup>™</sup> drivers
- Full ISAM functionality
- Powerful stand-alone index support

## Multiple-Platform Support

Mac OS, Mac OSX, MkLinux, Linux (PPC, Intel, SPARC, Alpha), Windows 95/98/ME/2000/NT, Novell Netware, Solaris (SPARC, Intel), Sun OS, OS/2, AIX, HP UX, SCO UnixWare, Interactive, AT&T Sys V, QNX, 88OPEN, FreeBSD, Lynx, Banyan Vines, and more...

**Supports ADSP,  
SPX, TCP/IP**



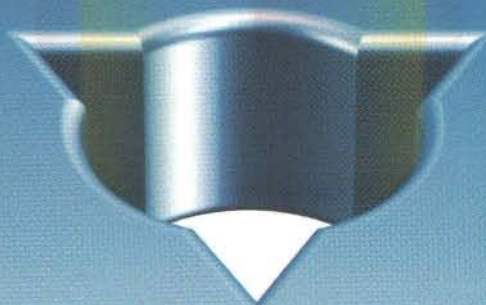
# FairCom<sup>®</sup>

QUALITY • THROUGH • EXPERIENCE

[www.faircom.com/maca](http://www.faircom.com/maca)

### FairCom Offices

|        |                  |
|--------|------------------|
| USA    | 573.445.6833     |
| EUROPE | +39.035.773.464  |
| JAPAN  | +81.59.229.7504  |
| BRAZIL | +55.11.3872.9802 |



© 2000 FairCom Corporation. Other company and product names are registered trademarks or trademarks of their respective owners.

**Mac Support Since 1985 • USA. 800.234.8180 • [info@faircom.com](mailto:info@faircom.com)**

```

if (!_FTakeStep(inodeNext, cactMax, paact, &cact, &inodeCur,
    pdist))
    goto LCleanup;

if (*pdist > distStop)
{
    // fix the object counts
    for (long inodeClean = 0; inodeClean < s_cnode;
        ++inodeClean)
        s_pdnnode[inodeClean].cobj = 0;
    goto LGiveUp;
}

// set the actual ordering used
long itskOrder, inodeOrder;
if (paitskOrderUsed != NULL)
{
    for (itskOrder = 0; itskOrder < s_ctsk; ++itskOrder)
    {
        TSK *ptskSetOrder = &s_ptask[itskOrder];

        if (ptskSetOrder->rankDrop == -1)
            ptaskSetOrder->rankDrop = ctskOrderDrop++;
        paitskOrderUsed[ptskSetOrder->rankDrop] = itskOrder;
    }
    for (inodeOrder = 0; inodeOrder < s_cnode; ++inodeOrder)
    {
        ENODE *penode = &s_pdnnode[inodeOrder];
        if (penode->rankGrab == -1)
            penode->rankGrab = cnodeOrderGrab++;
        paitskOrderUsed[s_ctsk+penode->rankGrab] = inodeOrder;
    }
}

LGiveUp:
fSuccess = fTrue;

LCleanup:
if (!fSuccess)
    *pdist = distMax;
return cact;
}

```

```

static long _InodeGetTarget(long inodeCur, long cobjCarry,
    int fUseWantedFlag, long *paitskOrder,
    long *pcnodeOrderGrab, long *pctskOrderDrop)
{
    TSK *ptsk;
    // if we have a chosen order, use that
    if (paitskOrder != NULL)
    {
        if (cobjCarry == 0)
        {
            long inodeGrab = -1;
            long rankGrab = s_cnode;
            for (long inode = 0; inode < s_cnode; ++inode)
            {
                ENODE *penode = &s_pdnnode[inode];
                if (penode->cobj == 0)
                    continue;

                if (penode->rankGrab < rankGrab)
                {
                    rankGrab = penode->rankGrab;
                    inodeGrab = inode;
                }
            }
            return inodeGrab;
        }
        else
        {
            TSK *ptskChoose = NULL;
            long rankChoose = s_ctsk;
            for (ptsk = s_ptskFirst; ptsk != NULL; ptsk = ptsk->ptskNext)
            {
                int fCarried = fUseWantedFlag ? ptsk->fWanted :
                    ptsk->inodeCur == -1;

                // only consider carried objects
                if (!fCarried)
                    continue;

                // pick the one with the lowest rank
                if (ptsk->rankDrop < rankChoose)
                {
                    rankChoose = ptsk->rankDrop;
                    ptskChoose = ptsk;
                }
            }
            return ptskChoose->inodeDst;
        }
    }

    // otherwise if we're not carrying anything, go to the nearest node with an object
    _ComputeNodeDistances(inodeCur);
    if (cobjCarry == 0)
    {
        // pick the nearest node with a task
        long inodeTarget = _InodeNearestTask(inodeCur);

        // try to assign a priority to this node
        if (s_pdnnode[inodeTarget].rankGrab == -1)
            s_pdnnode[inodeTarget].rankGrab = (*pcnodeOrderGrab)++;
        return inodeTarget;
    }

    // otherwise deliver the object with the nearest destination
    TSK *ptskDeliver = NULL;
    long distBest = distMax;
    for (ptsk = s_ptskFirst; ptsk != NULL; ptsk = ptsk->ptskNext)
    {
        if (fUseWantedFlag ? !ptsk->fWanted : ptsk->inodeCur != -1)
            continue;
        long distLook = _DistNodeToNode(inodeCur, ptsk->inodeDst);
        if (distLook < distBest)
        {
            distBest = distLook;
            ptskDeliver = ptsk;
        }
    }
    if (ptskDeliver->rankDrop == -1)
        ptskDeliver->rankDrop = (*pctskOrderDrop)++;
    return ptskDeliver->inodeDst;
}

```

<http://www.scientific.com>

## Professional Software Developers

Looking for career opportunities?

Check out our website!

Nationwide Service  
Employment Assistance  
Resume Help

Marketability Assessment

Never a fee

**Scientific Placement, Inc.**

800-231-5920 800-757-9003 (Fax)

das@scientific.com

```

#pragma mark -
// annealing functions

RandomizeOrder
static void _RandomizeOrder(long iTry, long cTry, long ctsk,
    long cnode, const long paitskOrderOld[],
    long paitskOrderNew[])
{
    // copy the old to the new
    BlockMove(paitskOrderOld, paitskOrderNew,
        sizeof(paitskOrderNew[0])*(ctsk*cnode));
    // swap some pairs of elements
    long cpairSwap = 1 + (ctsk + cnode) * (cTry-iTry)/(4*cTry);
    while (cpairSwap-- >= 0)
    {
        long itsk1, itsk2;
        // swap tasks or nodes?
        if (URand(ctsk + cnode) < ctsk)
        {
            // swap tasks
            itsk1 = URand(ctsk);
            itsk2 = URand(ctsk-1);
        }
        else
        {
            // swap nodes
            itsk1 = ctsk + URand(cnode);
            itsk2 = ctsk + URand(cnode-1);
        }
        if (itsk2 >= itsk1)
            ++itsk2;

        long itskT = paitskOrderNew[itsk1];
        paitskOrderNew[itsk1] = paitskOrderNew[itsk2];
        paitskOrderNew[itsk2] = itskT;
    }
    // utilities for translation to inode problem
}
#pragma mark -

```

```

NodeFromNode
// find the index to the specified Node
static long _InodeFromNode(Node node)
{
    long iinodeMin, iinodeLast;

    iinodeMin = 0;
    iinodeLast = s_cnode - 1;

    while (iinodeMin < iinodeLast)
    {
        long iinodeMid = (iinodeMin + iinodeLast + 1)/2;

        if (s_pdnnode[iinodeMid].node > node)
            iinodeLast = iinodeMid - 1;
        else
            iinodeMin = iinodeMid;
    }

    return s_pdnnode[iinodeMin].inode;
}

```

```

CmpEnodeNode
// compare two ENODE entries by their node values for HeapSort
static int _CmpEnodeNode(INDEX i1, INDEX i2, void *pv)
{
    ENODE *pdnnode = (ENODE *)pv;

    return pdnnode[i1].node - pdnnode[i2].node;
}
// compare two ENODE entries by their inode values for HeapSort
static int _CmpEnodeInode(INDEX i1, INDEX i2, void *pv)
{
    ENODE *pdnnode = (ENODE *)pv;

    return pdnnode[i1].inode - pdnnode[i2].inode;
}
#pragma mark -

```

```

FlakeStep
static int _FTakeStep(long inodeDst, long cactMax,
    Action paact[], long *pact, long *pinodeCur, long
    *pdist)

```

```

{
    if (*pact >= cactMax)
    {
        printf("OOM in _FTakeStep\n");
        return fFalse;
    }

    long dist;
    if (!_FConnectedNodes(*pinodeCur, inodeDst, &dist))
        return fFalse;

    *pdist += dist;
    *pinodeCur = inodeDst;
    Action *pact = &paact[(*pact)++];
    pact->action = kMoveTo;
    pact->node = s_pnode[inodeDst];
    return fTrue;
}

```

```

GrabObject
static int _FGrabObject(long cactMax, Action paact[],
    long *pact, long *pwgtCarry, long *pcobjCarry,
    long inode, long itsk)
{
    if (*pact >= cactMax)
    {
        printf("OOM in _FGrabObject\n");
        return fFalse;
    }

    Action *pact = &paact[(*pact)++];
    TSK *ptsk = &s_patsk[itsk];

    pact->action = kPickUpObject;
    pact->object = ptask->itaskObject;

    ptask->inodeCur = -1;
    s_pdnnode[inode].cobj -= 1;

    ++(*pcobjCarry);
    *pwgtCarry += ptask->wgt;
}

```

## Felt Tip Sound Studio



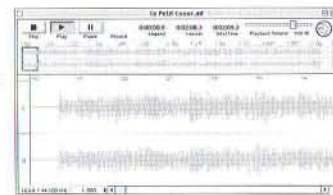
### Record and Edit Audio

with a sound editor designed for the Mac.

Sound Studio will allow you to make quick edits with an interface as easy as a text editor. Add polish to recordings with fades, normalization, and edits. Create your own mixes. Transform them with effects.

#### Sound Studio features

- up to 16 bits, 2 channels, and 65 kHz
- up to 2 GB of audio
- AIFF, Sound Designer 2, WAVE, System 7 Sound, and QuickTime import
- edit with sample accuracy
- fade, amplify, normalize, and invert
- delay, echo, reverse, and swap channels
- smooth and emphasize
- resample and pitch shift
- snap to zero crossings, snap to grid



**\$35**  
Sound Studio

download free 14-day trial  
or order online at  
**www.felttip.com**

QuickTime is a trademark, used under license.
price in US dollars

Felt Tip Software, 807 Keely Place, Philadelphia PA 19128-2326, USA

```

    return fTrue;
}

static int _FDropObject(long cactMax, Action paact[],
    long *pact, long *pwgtCarry, long *pobjCarry,
    long inode, long itsk)
{
    if (*pact >= cactMax)
    {
        printf("OOM in _FDropObject\n");
        return fFalse;
    }

    Action *pact = &paact[(*pact)++];
    TSK *ptsk = &s_ptsk[itsk];

    pact->action = kDropOffObject;
    pact->object = ptask->itaskObject;

    ptask->inodeCur = inode;
    if (ptsk->inodeDst != inode)
        s_pdnnode[inode].cobj += 1;

    -(*pobjCarry);
    *pwgtCarry -= ptask->wgt;

    return fTrue;
}

#pragma mark -

static void _MapConnections(long cconn, const Connection
    paconnection[])
{
    s_cconn = cconn;

    try

```

## BasePlant

200 classes to enhance PowerPlant™

- AppleEvents
- AppleScript
- Collection Manager
- Data Browser
- Disk Item Filter
- Folder Parser
- Icon Services
- Multiprocessing
- Preferences Manager
- Starting Point
- Window Proxy

And Much More!

\$149 For commercial, in-house development  
 \$35 For personal freeware and shareware

CAFEDERIC.COM

```

    s_paconn = new CONN[cconn];

    Assert(s_pacmap == NULL);
    s_pacmap = new CMAP[s_cnode*s_cnode];
    for (long iT = s_cnode*s_cnode; -iT >= 0; )
    {
        s_pacmap[iT].distDirect = -1;
        s_pacmap[iT].distPath = -1;
    }

    s_panodec = new NODC[s_cnode];
}
catch(...)
{
    printf("Bob, I need more memory, somewhere
        around %ld bytes\n",
        sizeof(CONN)*cconn
        +sizeof(CMAP)*s_cnode*s_cnode
        +sizeof(NODC)*s_cnode
        );
}

for (long iconn = 0; iconn < cconn; ++iconn)
{
    CONN *pconn = &s_paconn[iconn];
    const Connection *pconnection = &paconnection[iconn];

    pconn->dist = pconnection->distance;
    pconn->inode1 = _InodeFromNode(pconnection->node1);
    pconn->inode2 = _InodeFromNode(pconnection->node2);

    s_pacmap[pconn->inode1*s_cnode + pconn->
        >inode2].distDirect =
        pconn->dist;
    s_pacmap[pconn->inode2*s_cnode + pconn->
        >inode1].distDirect =
        pconn->dist;
}

static void _FreeConnectionMap()
{
    if (s_paconn != NULL)
    {
        delete s_paconn;
        s_paconn = NULL;
    }
    if (s_pacmap != NULL)
    {
        delete s_pacmap;
        s_pacmap = NULL;
    }
    if (s_panodec != NULL)
    {
        delete s_panodec;
        s_panodec = NULL;
    }
}

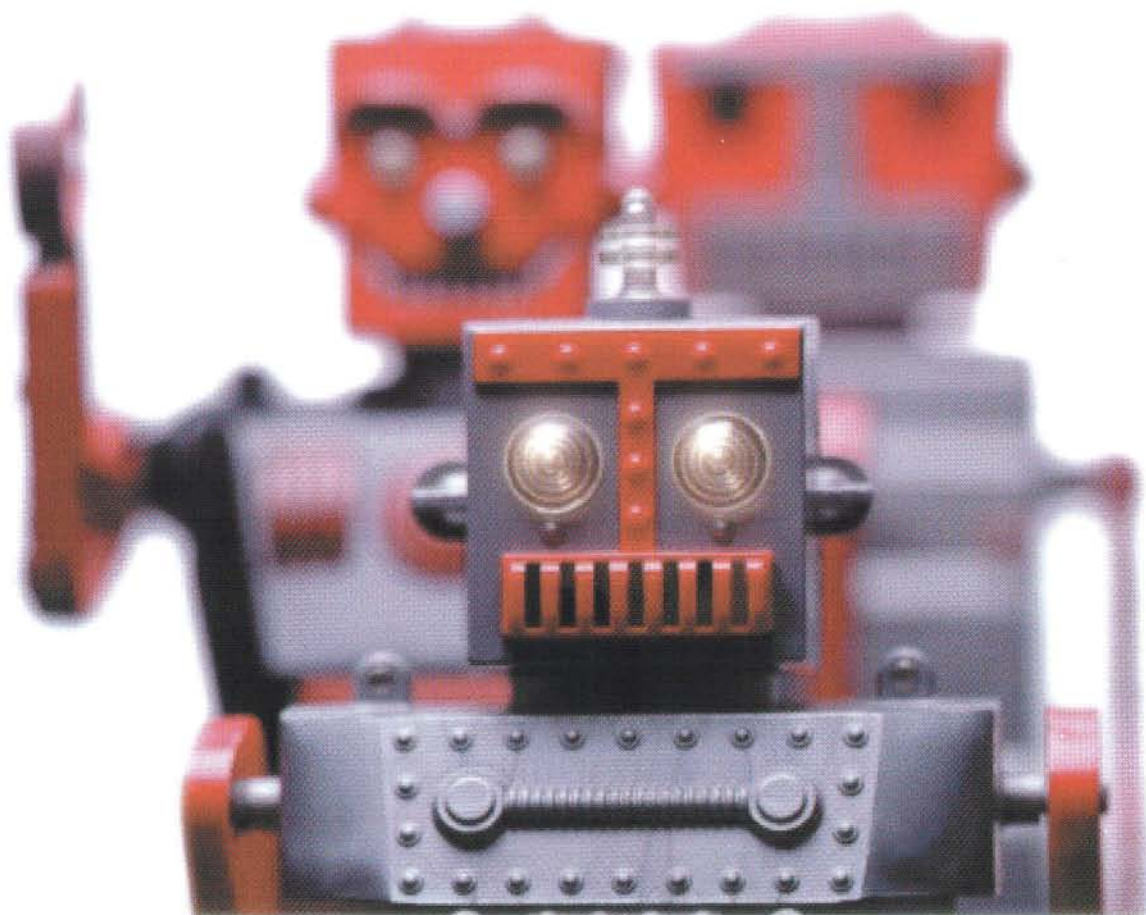
static int _FConnectedNodes(long inode1, long inode2,
    long *pdist)
{
    long distT;
    if (pdist == NULL)
        pdist = &distT;
    *pdist = s_pacmap[inode1*s_cnode + inode2].distDirect;
    return *pdist != -1;
}

static void _ComputeNodeDistances(long inode)
{
    if (s_pacmap[inode*s_cnode + inode].distPath == -1)
        _CnodeFindMinPath(inode == 0 ? 1 : 0, inode, NULL);
}

static long _CnodeFindMinPath(long inodeStart, long inodeEnd,
    long painodePath[])
{
    if (inodeStart == inodeEnd)

```

# A NEW SYSTEM IS COMING.



## NETOCTOPUS IS READY. ARE YOU?

Mac OS X is coming. It's right around the corner. You may be going through your preparation checklist right now, wondering if you have enough time:

- ✓ Inventory all installed software
- ✓ Check software versions
- ✓ Check hard disks and memory on all computers
- ✓ Identify hardware upgrade needs
- ✓ Print summary reports

With netOctopus, you can do all of that with a few mouse clicks. Then, when OS X arrives, get the new version of netOctopus, and use it to distribute all of your new software throughout your network seamlessly.



Timbuktu Pro, the #1 remote control and file transfer software for the Mac, integrates seamlessly into netOctopus, for even greater network support!

**With netOctopus, you would practically be ready for OS X today.  
Without netOctopus, you may be running out of time.**



[www.netoctopus.com](http://www.netoctopus.com)

```

    return 0;
// clear the NODC array
for (long inode = 0; inode < s_cnode; ++inode)
{
    s_panode[inode].distFromStart = distMax;
    s_panode[inode].fCheckNeighbors = fFalse;
}

// seed the fill with the end node
s_panode[inodeEnd].distFromStart = 0;
s_panode[inodeEnd].fCheckNeighbors = fTrue;

long distPathKnown = s_pacmap[inodeStart*s_cnode +
    inodeEnd].distPath;

// expand out from base node until we've exhausted all possibilities
for (int fExpanded = fTrue; fExpanded;)
{
    fExpanded = fFalse;

    for (long inodeSrc = 0; inodeSrc < s_cnode; ++inodeSrc)
    {
        if (!s_panode[inodeSrc].fCheckNeighbors)
            continue;
        for (long inodeDst = 0; inodeDst < s_cnode; ++inodeDst)
        {
            long dist;
            if (inodeSrc == inodeDst ||
                !_FConnectedNodes(inodeSrc, inodeDst, &dist))
                continue;
            if (s_panode[inodeSrc].distFromStart + dist <
                s_panode[inodeDst].distFromStart)
            {
                s_panode[inodeDst].distFromStart =
                    s_panode[inodeSrc].distFromStart + dist;
                s_panode[inodeDst].fCheckNeighbors = fTrue;

                if (inodeDst == inodeStart && distPathKnown != -1)
                    if (s_panode[inodeDst].distFromStart ==
                        distPathKnown)
                        goto LHavePath;
            }
        }
    }
}

```

```

        fExpanded = fTrue;
    }
    s_panode[inodeSrc].fCheckNeighbors = fFalse;
}

// if we need to, record the results for this node
if (s_pacmap[inodeEnd*s_cnode + inodeEnd].distPath == -1)
{
    for (long inodeConnect = 0; inodeConnect < s_cnode;
        ++inodeConnect)
    {
        long dist = s_panode[inodeConnect].distFromStart;
        s_pacmap[inodeConnect*s_cnode + inodeEnd].distPath =
            dist;
    }
    s_pacmap[inodeEnd*s_cnode + inodeEnd].distPath = 0;
}

// walk from the start node to the end node
LHavePath:
if (s_panode[inodeStart].distFromStart == distMax)
{
    printf("Hey! There's no way to get from node %ld to node
        %ld!\n", s_panode[inodeStart], s_panode[inodeEnd]);
    return 0;
}

long cnodePath;
long inodeCur;
for (cnodePath = 0, inodeCur = inodeStart;
    inodeCur != inodeEnd; ++cnodePath)
{
    long inodeNext;
    long distConnect;
    long distCur = s_panode[inodeCur].distFromStart;
    for (inodeNext = 0; inodeNext < s_cnode; ++inodeNext)
        if (_FConnectedNodes(inodeNext, inodeCur,
            &distConnect) &&
            s_panode[inodeNext].distFromStart + distConnect ==
            s_panode[inodeCur].distFromStart)
            break;

    if (painodePath != NULL)
        painodePath[cnodePath] = inodeNext;
    inodeCur = inodeNext;
}

return cnodePath;
}

```

```

static long _InodeNearestTask(long inodeCur)
{
    // find the nearest object
    long distBest = distMax;
    long inodeBest = -1;
    for (long inodeLook = 0; inodeLook < s_cnode; ++inodeLook)
    {
        if (s_pdnnode[inodeLook].cobj == 0)
            continue;

        long dist = _DistNodeToNode(inodeCur, inodeLook);
        if (dist < distBest)
        {
            inodeBest = inodeLook;
            distBest = dist;
        }
    }

    return inodeBest;
}

#pragma mark -

static void _ResetState()
{
    // fill in object counts in dnode
    for (long itsk = 0; itsk < s_ctsk; ++itsk)
    {
        TSK *ptsk = &s_patsk[itsk];
    }
}

```

**Valentina**  
object-relational database engine



**The fastest database engine  
for the MacOS**

**It operates 100's and sometimes a 1000  
times faster than other systems**

|                                              |                  |
|----------------------------------------------|------------------|
| <b>Valentina (AppleScript) (68K/PPC/X)</b>   | <b>\$49</b>      |
| <b>Valentina C++ SDK (68K/PPC/X/Win32)</b>   | <b>\$499/699</b> |
| <b>Valentina for Java (PPC/X/Win32)</b>      | <b>\$299</b>     |
| <b>Valentina for REALbasic (PPC/X/Win32)</b> | <b>\$199/299</b> |
| <b>Valentina for Director (PPC/Win32)</b>    | <b>\$199/299</b> |
| <b>Valentina XCMD (PPC/Win32)</b>            | <b>\$199/299</b> |
| <b>Valentina for WebSiphon (PPC)</b>         | <b>\$299</b>     |

**Make your application's database operations blazingly fast!**

Order Directly **www.paradigmasoft.com**  
from Our Web Site Hosted by MacServe.net

**Download full featured evaluation version**

```

    ptsk->inodeCur = ptsk->inodeStart;
    if (ptsk->inodeCur != ptsk->inodeDst)
        s_pdnnode[ptsk->inodeCur].cobj += 1;

    // chain the records together
    ptsk->ptskNext = ptsk+1;
}

// set up the pointers at the start and end of the list
s_ptskFirst = s_patsk;
s_patsk[s_ctsk-1].ptskNext = NULL;
}

static TSK *_PtskRemoveTsk(TSK *ptsk)
{
    TSK **pptsk = &s_ptskFirst;
    while ((*pptsk) != ptsk)
        pptsk = &(*pptsk)->ptskNext;

    TSK *ptskNext = ptsk->ptskNext;
    ptsk->ptskNext = NULL;
    *pptsk = ptkNext;
    return ptkNext;
}

#pragma mark -

```

```

// return a random number in the range [0,uMax)
// NOTE: this only works for values that fit into a short
static unsigned short URand(unsigned short uMac)
{
    unsigned short u, uLim;

    // remove bias toward small numbers
#define wRandMax 0xFFFF
    Assert(sizeof(int) > sizeof(short));
    Assert(uMax < wRandMax);
    Assert(wRandMax > 0 && (wRandMax+1) > wRandMax);
    uLim = wRandMax - ((wRandMax+1) % uMac);
    do
    {
        u = (unsigned short)Random();
    } while (u > uLim);

    return u % uMac;
}

```

// heap sort code

```

static void _Swap(INDEX i1, INDEX i2, void *pv, int cbFoo);

void HeapSort(void *prgfoo, INDEX iMac, int cbFoo, PFNCMP
pfncmp)
{
    INDEX iTop, iLast, iUnder, iCur;

    if (iMac == 0)
        return;

    iTop = iMac/2; // first initial line worker */
    iLast = iMac - 1; // the last element */

    for(;;)
    {
        if (iTop > 0)
        {
            // hire a new supervisor, put him in his place */
            -iTop;
        }
        else
        {
            // retire the chairman of the board,
            // trickle the last line worker into his correct spot
            // */
            _Swap(iLast, 0, prgfoo, cbFoo);

            if (iLast- <= 1)
                return;
        }
    }
}

```

```

iCur = iTop; // new supervisor */
iUnder = iCur + iCur + 1; // first underling */

while (iUnder <= iLast)
{
    // compare to the better underling */
    if (iUnder < iLast)
        if ((*pfncmp)(iUnder, iUnder+1, prgfoo) < 0)
            ++iUnder;
    if ((*pfncmp)(iCur, iUnder, prgfoo) < 0)
    {
        // promote the underling */
        _Swap(iCur, iUnder, prgfoo, cbFoo);

        // check the demoted "supervisor" against his new underlings */
        iCur = iUnder;
        iUnder += iUnder + 1; // first underling of demoted supervisor */
    }
    else
    {
        // we're done sifting, terminate loop */
        break;
    }
}
}
}
}

```

```

static void _Swap(INDEX i1, INDEX i2, void *pv, int cbFoo)
{
    char *pb1, *pb2, bT;
    int cb;
    pb1 = (char*)pv + (long)i1*cbFoo;
    pb2 = (char*)pv + (long)i2*cbFoo;

    for (cb = cbFoo; cb-; )
    {
        bT = *pb1;
        *pb1++ = *pb2;
        *pb2++ = bT;
    }
}

```

MT

# MacWise

The best in Macintosh  
terminal emulation

Version 4.26

Emulations Include  
Wyse 50, 60, and 370  
Viewpoint, VT100, Esprit, TV925  
Color and Fkeys supported

**Carnation  
Software**  
512•858•9234

[www.carnation-software.com](http://www.carnation-software.com)

# The products you need, with the prices and service you deserve... guaranteed.

## Power Tools for Programmers!

### CodeWarrior Pro 6

**NEW!**

CodeWarrior for Mac OS is a powerful integrated development environment that includes a fully object-oriented application development framework called PowerPlant. With CodeWarrior for Mac OS you can quickly develop reliable, professional quality applications that execute on Classic Mac OS, or OS X.

**NEW Version**



ONLY! **\$359**

### REALbasic 2.1

**NEW!**

REALbasic is the programming tool for "the rest of us." Each window type, control, and menu is preconfigured and instantly works as it should. The drag and drop Window Editor allows you to quickly and easily create your application's interface so you can focus on the important part—your creativity. Includes 900 page manual, examples and tutorial on CD ROM! The professional version has all the power of the standard, plus database support and allows you to cross compile your code for Windows with a single click!

As low as

**\$139**



### Spotlight

Spotlight is the first Macintosh "Automatic Debugger". It can automatically locate run time errors in your code and display the offending source code line. Unlike similar tools on other platforms Spotlight is easy to use. No source code changes are necessary for application debugging. Spotlight can automatically check for wild pointers, memory leaks, overwrites, underwrites, invalid dereferencing of handles, and even toolbox parameter validity checking -- spotlight knows Macintosh verifying parameters to over 400 toolbox calls.



**\$189**

### VOODOO Server

VOODOO Server is a version control system for software developers using Metrowerks CodeWarrior under Mac OS. VODOO Server and the corresponding VODOO clients (the included CodeWarrior VCS plug-in and the VODOO Admin application) are designed to offer reliable and robust version control features while minimizing the administrative overhead that usually accompanies version control. If you're a single programmer or managing a team of developers, version control can make or break your project. Do it right, with VODOO



**\$79**

### Resorcerer 2.2

Resorcerer is the only supported general-purpose resource editor for Macintosh. Relied upon by thousands of Mac developers, Resorcerer features a wealth of powerful yet easy-to-use tools for easier, faster, and safer editing of Macintosh data files and resources. Whether you have to parse a picture, debug a data fork, design and try out Balloon Help, create a scripting dictionary, create anti-aliased icons, design and edit a custom resource with 40,000 fields in it, create C source code to run a dialog, or any of hundreds of other resource-related tasks, Resorcerer's magic will quickly save you time and money.



**\$239**

### Future BASIC 3

**NEW!**

One of the most flexible and powerful development environments on the Macintosh today! Easily create programs with the visual program editor, drop into the BASIC editor to define powerful logic with the worlds easiest programming language, or work directly with the Macintosh toolbox. The only BASIC compiler on the market that gives you 100% access to all of the power of the Macintosh toolbox!



**\$159**

and hundreds more!

|                                                |                                             |                                                |                                              |                                             |                                                  |                                                   |                                       |                                                |
|------------------------------------------------|---------------------------------------------|------------------------------------------------|----------------------------------------------|---------------------------------------------|--------------------------------------------------|---------------------------------------------------|---------------------------------------|------------------------------------------------|
| <p><b>PageCharmer 2.0</b><br/><b>\$139</b></p> | <p><b>Scripter 2.0</b><br/><b>\$179</b></p> | <p><b>Script Debugger</b><br/><b>\$189</b></p> | <p><b>ToolsPlus Lite</b><br/><b>\$99</b></p> | <p><b>FaceSpan 3.0</b><br/><b>\$179</b></p> | <p><b>WebSpice 1,000,000</b><br/><b>\$89</b></p> | <p><b>WebSpice Animations</b><br/><b>\$89</b></p> | <p><b>MkLinux</b><br/><b>\$39</b></p> | <p><b>PowerKey Rebound</b><br/><b>\$89</b></p> |
|------------------------------------------------|---------------------------------------------|------------------------------------------------|----------------------------------------------|---------------------------------------------|--------------------------------------------------|---------------------------------------------------|---------------------------------------|------------------------------------------------|

# Developer DEPOT®

PO Box 5200 • Westlake Village, CA • 91359-5200 • Voice: 800/MACDEV-1 (800/622-3381)  
Outside US/Canada: 805/494-9797 • Fax: 805/494-9798 • E-mail: orders@devdepot.com

**www.devdepot.com**

**Master the Web!**

## WebSTAR Server Suite 4.3

WebSTAR Server Suite is a complete set of powerful and easy-to-use Internet servers for the Mac OS. Effortlessly serve web pages, host email accounts, publish databases, and share files - all with a single application on one Mac! WebSTAR Server Suite is perfect for Internet or Intranet serving, single or multiple sites, small and large businesses - it's power and ease-of-use saves any organization time and money.



**\$539**

## CyberGauge 3.0

Monitor the bandwidth usage of up to five different machines on your network! Do you need to upgrade your webserver? How hard is your eMail server working? Are you getting all the bandwidth you're paying for? Not only can CyberGauge answer all these questions, new features allow CyberGauge to eMail or page your network device becomes unresponsive or passes a threshold of usage you define - an essential first line of defense for early detection of denial of service attacks and necessity for warning you and tracking quality of ISPs that may have brown outs and shutdowns.



**\$249**

## Funnel Web Pro

Funnel Web is the ultimate web analysis solution for professionals. Specifically designed for profiling web site usage and monitoring customer usage patterns, Funnel Web is ideal for examining server performance and online effectiveness. Funnel Web can analyze log file formats from any server, WebSTAR, WebTen, even Unix or NT hosted servers. Discover the most popular pages on your site, track server loads & optimize server performance, profile visitors based on organization, domain name, country, browser, etc. Funnel Web does it all!



As low as

**\$329**

## NetBarrier

NetBarrier offers a Personal Firewall, Antivandal protection, and Internet Filtering. Protect your machine from intrusions by Internet or AppleTalk. Incorrect passwords and individual actions are logged, you are alerted to hostile actions, and intruders are easily isolated. Internet Filtering allows you to be sure that passwords, credit card numbers, and other sensitive information can never be exported from your computer - the content itself is filtered before any transfer! (Rated 4 mice by Macworld Magazine)



**\$49<sup>95</sup>**

**...and great hardware solutions!**

## 2 USB PCI Card

Add two USB ports to your older Macintosh. Connect up to 127 devices to the Universal Serial Bus (USB) that is Apple's new standard for desktop connectivity. USB mouse devices, keyboards, joysticks, game controllers, printers, scanners - connect them all to your current computer. Installs in minutes!



**\$32<sup>95</sup>**

## Macsense USB Full Size Keyboard

Just plug this keyboard into your Mac and start typing! The UKB-600 keyboard from Macsense is designed to get you typing quickly and easily, without any hassle or compatibility worries. It features two tone translucent design, colored to match your favorite flavor of Macintosh. It offers soft touch with positive tactile feedback and build built in USB port on either side of the keyboard. Includes a 5' USB cable and is 100% Macintosh compatible, simply plug and play, as easy as Macintosh!



**\$44<sup>95</sup>**

## Dr. Bott Moni Switch ADB or USB

Do you need 4 monitors and 4 keyboards for your 4 servers? With Dr. Bott Moni-Switch you can connect a single keyboard and monitor to up to 4 machines at once! A simple flick of a switch directs the video input and keyboard commands to the appropriate CPU! Available in USB and ADB models, with 2 or 4 machine support, and bundles with USB PCI cards so you can mix and match USB and ADB machines with the same Moni-Switch! Great for programmers to do back ground compiles, ideal for server rooms overcrowded with monitors and keyboards!



As low as

**\$129<sup>95</sup>**

## Macsense Internet Sharing Router

Looking to get your whole office online without shelling out thousands of dollars? If so, the XRouter Internet Sharing Hub offers the perfect solution. This amazing Ethernet-to-Ethernet hub connects an entire network of up to 252 users to the Internet using only one ISP account and one Cable or DSL modem!



**\$199**

By Tim Monroe

# A Goofy Movie

## Working with Sprites in QuickTime Movies

### INTRODUCTION

The major new technology introduced in QuickTime version 2.1 (released in 1995) was support for sprites and sprite animation. A *sprite* is a graphical object that has a number of properties, including its current image, location, size, layer, graphics mode, and visibility state. These properties determine the appearance of the sprite at any instant in time. By varying one or more of these properties over time, we can *animate* the sprite. For instance, **Figure 1** shows the first frame of a QuickTime movie that contains one sprite, whose image is the original icon for the QuickTime system extension.



**Figure 1:** A movie with a sprite

We can move the icon to the right by gradually changing the horizontal location of the sprite. In addition, we can change the image associated with the sprite at any time. In this case, we'll change the icon from the old QuickTime extension icon to the new QuickTime extension icon when the sprite gets to the halfway point, as shown in **Figure 2**. The new icon then continues moving at the same rate until it reaches the right side of the movie box.

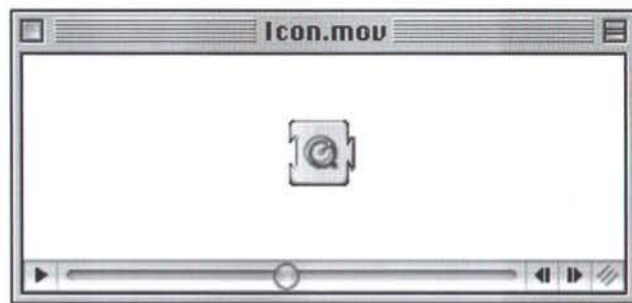


**Figure 2:** The sprite with a new image

**Figure 3** shows another sprite movie. In this case, there is only one image associated with the sprite for the entire duration of the movie, and the sprite location remains constant. We'll animate this sprite by changing the graphics mode, from totally transparent to totally opaque. What we've done here is recreate, using sprite animation, the appearing-penguin movie that was the very first QuickTime movie we built in this series of articles (see "Making Movies" in *MacTech*, June 2000).

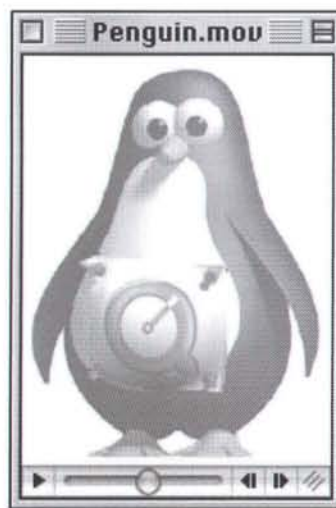


**Figure 3:** The penguin movie using sprite animation

**Tim Monroe** is amazed at how fast his new lizard *Libra* is growing. Maybe he'll try feeding insects to his own children to see if they grow any faster. In the meantime, you can contact him at [monroe@apple.com](mailto:monroe@apple.com).

Recall how we went about creating the first version of our penguin movie: we opened a picture resource and drew that picture into an offscreen graphics world; we compressed the data in that graphics world and added it as a frame to the movie file. Then we repeated the drawing and compressing 99 times, each time with gradually more opacity, to create 99 additional movie frames. The resulting movie file contained 100 compressed images, for a total size of about 472 kilobytes.

Using sprite animation, we can reduce that size dramatically. Our new penguin movie file contains only a single compressed image and 100 sets of "instructions" that indicate the desired level of opacity in each movie frame. The 100 movie frames are generated *at playback time* by the sprite media handler from that image and those 100 sets of instructions. The size of the sprite version of the penguin movie is only about 36 kilobytes. (No, that's no typo; the exact same visual output can be achieved with a file less than one tenth the size of our original movie file.)

Already you can see that sprite animation differs significantly from what's often called *cel animation*, where each frame of the animation is a fully-rendered picture of some characters superimposed on a background image. If cel animation is like a recorded symphony, then sprite animation is more like a set of sounds together with instructions for playing back those sounds in the right order and at the right time. If the instructions in a sprite movie aren't too

complicated, they can be executed at runtime just as smoothly as decompressing and playing back the fully-rendered version of the movie. And, as we've seen, the images and instructions can take up a lot less space.

QuickTime 3.0 extended the capabilities of the sprite media handler by adding support for *wired sprites*, or sprites that react to mouse events (and other kinds of events) and that have various actions attached (or "wired") to them. For instance, we can use a wired sprite to control the properties of the sprites in the movie (so that clicking on one sprite causes another sprite to disappear or to change location). Or, we can control various aspects of movie playback, such as the volume and balance of a sound track, or the graphics mode of a video track. By using wired sprites in a movie, we can add a level of interactivity previously unavailable in QuickTime.

In this article, we're going to learn how to create movies that contain sprite tracks. We'll see how to create both the icon movie shown in **Figure 1** and the penguin movie shown in **Figure 3**. Our sample application this month is called QTSprites, and its Test menu is shown in **Figure 4**.

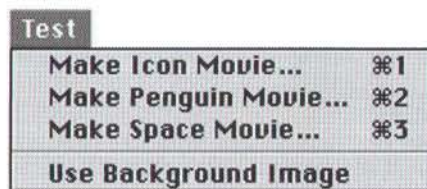


**Figure 4:** The Test menu of QTSprites

## Webmasters! Start your web hosting business with edition.net reseller hosting!

dedicated platforms  
single and multi-domain hosting  
your choice of Mac OS or Linux platforms  
FileMaker Pro hosting featuring Lasso  
QuickTime streaming server

Our web-based control panels make it easy to  
deploy your multi-domain server

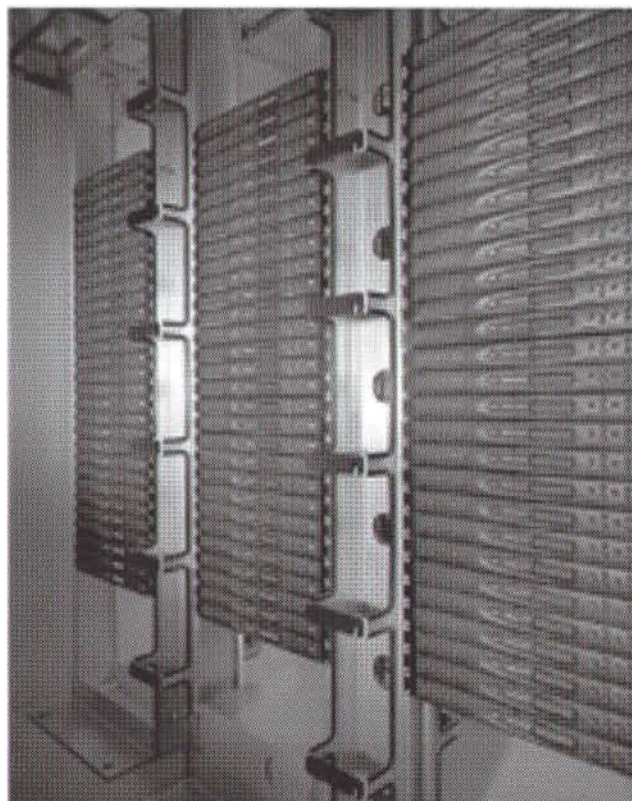
**edition.net – the Fine Arts Network**

<http://www.edition.net>

[info@edition.net](mailto:info@edition.net)

+1 (877) 225-3821 toll free

+1 (310) 379-8563 international



# Always Thinking

Professional Macintosh & Internet Development

**Always Thinking's** professional developers will help you meet your Macintosh and Internet deadlines! So if you're...

- on a tight deadline and need additional talent
- losing valuable development time debugging
- having trouble finding good developers

... **Always Thinking's** team of experienced programmers will provide you with a timely and affordable solution.

We deliver more than code — a complete project. Our software engineers work with you to:

- Create clear, solid project specifications
- Design and develop your application or web site
- Tune and optimize your software's performance
- Thoroughly test your application or site
- Completely document your project
- Provide training to your team

## Commercial Product Development

Do you have an exciting idea for an application? Turn to **Always Thinking** to make it a reality. We have firsthand experience developing and shipping award-winning commercial applications for our clients and our own Thinking Home, a 2000 Apple Design Award winner.

## Web Site Design & Development

Get a sound e-commerce system tailored for both your immediate needs and long-term growth. Our engineers can develop the Internet applications to transform your company into an e-business.

Successful web sites are more than graphics and code. We have the Internet marketing know-how to ensure your site is an effective business tool.

Realize substantial savings by moving to online pre-sales information, ordering and support.

Tell us about your project, toll-free

**(800) 556-9559**

(843) 986-0308



**Always Thinking, Inc.**  
27 James Byrnes Street  
Beaufort, SC 29902

[www.alwaysthinking.com](http://www.alwaysthinking.com) [sales@alwaysthinking.com](mailto:sales@alwaysthinking.com)

(The "space movie" is a more complex movie that does sprite animation using location changes, layer changes, and image changes. We won't learn how to build it in this article, but the code for doing so is contained in the file `QTSprites.c`.)

We'll begin by looking in more detail at sprites and their properties. Then we'll take a look at the structure of sprite tracks and see how to build the icon and penguin sprite movies. Toward the end, we'll see how to add some simple interactivity to our sprite movies without using wired sprites. We'll postpone our investigation of wired sprites to an upcoming *QuickTime Toolkit* article.

## SPRITE PROPERTIES

In a QuickTime movie, a sprite is a graphical object that belongs to a sprite track (of type `SpriteMediaType`). In the simplest case, the basic appearance of a sprite is set by selecting one out of an array of images associated with the sprite. The current *sprite image index* is one of the five main *sprite properties*, defined using these constants:

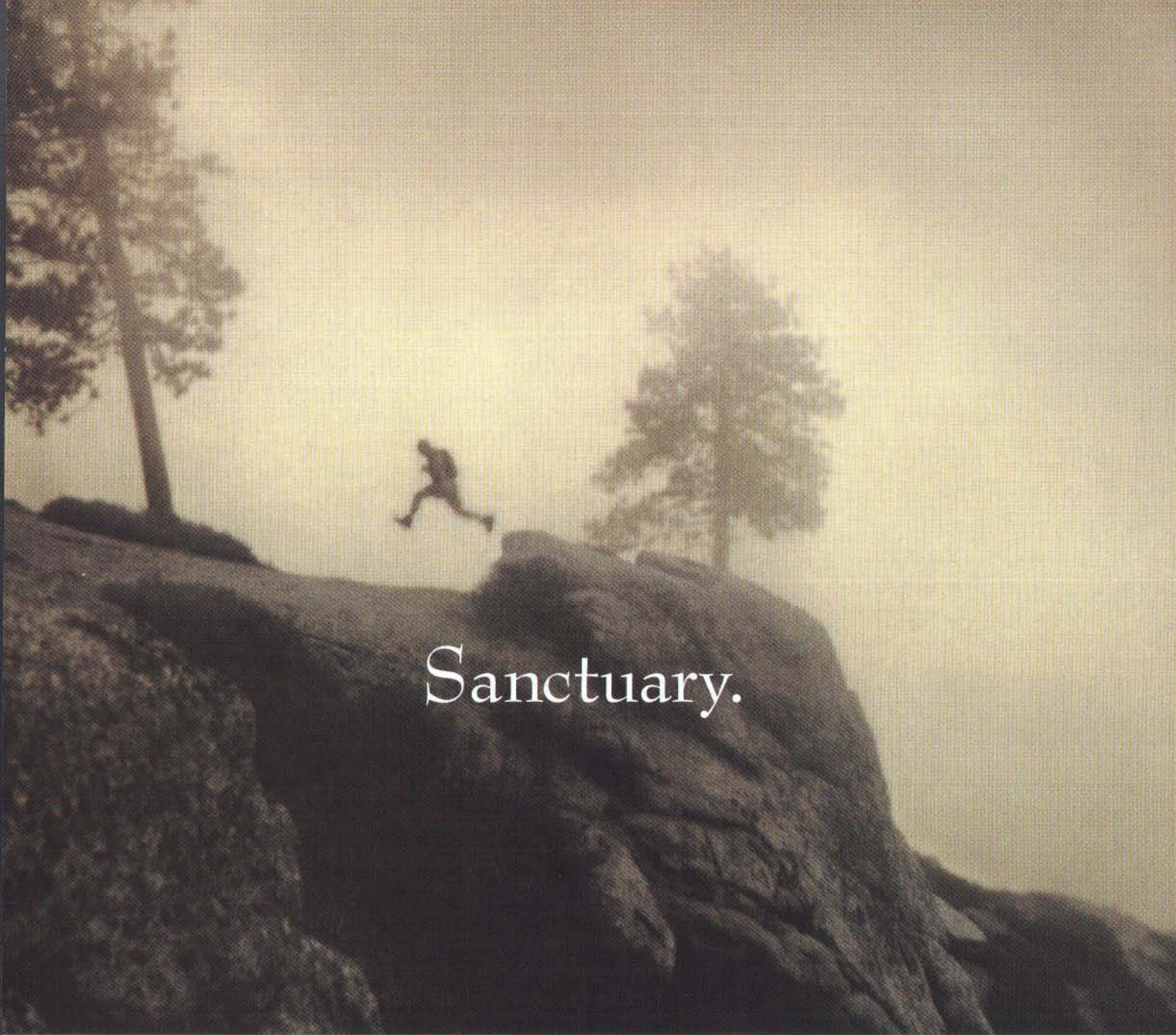
```
enum {
    kSpritePropertyMatrix          = 1,
    kSpritePropertyVisible         = 4,
    kSpritePropertyLayer           = 5,
    kSpritePropertyGraphicsMode    = 6,
    kSpritePropertyImageIndex      = 100
};
```

The *sprite matrix* is a 3-by-3 matrix that controls the location, size, and rotation of the sprite image within the sprite track. (A sprite's matrix is added to the track matrix of the sprite track.) The *sprite visibility state* is an integer value that controls whether the sprite is currently visible. (This value is interpreted as a Boolean value but is stored in the movie file as a 16-bit short integer.) The *sprite layer* is an integer value that determines, when two or more sprites have locations that overlap, which sprite is drawn on top of the other sprite(s). Sprites with lower layer values are drawn on top of sprites with higher layer values; if we want to ensure that some sprite is drawn behind all other overlapping sprites, we can set its layer to the special value `kBackgroundSpriteLayerNum` (appropriately defined in the file `Movies.h` as 32767). Finally, the *sprite graphics mode* determines how the sprite is drawn into the sprite track. We specify a sprite graphics mode using a structure of type `ModifierTrackGraphicsModeRecord`, defined like this:

```
struct ModifierTrackGraphicsModeRecord {
    long          graphicsMode;
    RGBColor      opColor;
};
```

This structure contains the QuickDraw graphics mode and the color used by some of those graphics modes. For instance, if `graphicsMode` is `blend`, then `opColor` specifies the weight color (which determines the amount of blending of the source and destination pixels).

A sprite's current image does not have to come from a sprite image array. Instead, the sprite media handler can use a video track in the same QuickTime movie as the source for the sprite's

A black and white photograph of a person jumping over a large, dark rock in a forest. The person is in mid-air, and the background shows trees and a hazy sky.

# Sanctuary.

## Move to **digital.forest**. The leading Internet Service Provider for the Macintosh community.

You need maximum flexibility, security and performance from your ISP. You'll find it with digital.forest. We're the proven provider of Macintosh Internet service.

Founded in 1994, our specialty is Mac systems and technologies. In fact, digital.forest is the world's largest Macintosh Hosting Provider. And our hard-earned knowledge of Mac server software is the broadest in the business.

We pioneered Macintosh Server Colocation and FileMaker Pro database hosting services. And our commitment to service keeps growing. So whether you choose us to host your files or house your servers, you've chosen the best.

Go with the industry leader. Discover the sanctuary of digital.forest.

**digital.forest**



**8 7 7 - 7 2 0 - 0 4 8 3**

[info@forest.net](mailto:info@forest.net)  
[www.forest.net](http://www.forest.net)

Call toll-free in the U.S. and Canada.  
For international inquiries, please call 425-483-0483.



**FileMaker**  
ASSOCIATE

FileMaker Pro is a registered trademark of FileMaker, Inc. The digital.forest logo is trademark or registered trademark of digital.forest, Inc.

© digital.forest, Inc. 2000. All rights reserved

image. This allows us to create even more intricate animations than are possible by simply varying the five basic sprite properties. As a very simple example, we could create a sprite track containing two sprites, one whose image looks like a television set and a superimposed sprite whose images are derived from a video track. The net effect would be a sprite track containing a television playing the video.

### SPRITE TRACKS

A sprite track consists of one or more sprite media samples. There are two basic kinds of sprite media samples: (1) those that define the sprite image array and set the initial properties of a frame, and (2) those that animate the sprites in the track by specifying changes to the sprites' properties. The sprite media handler relies on the distinction between key frames and difference frames, which we encountered in the previous article ("Honey, I Shrunk the Kids" in *MacTech*, February 2001). The image arrays and initial properties are stored in key frames, and the sprite property changes are stored in difference frames. The only departure here is purely terminological: when we're working with sprite data, the difference frames are called *override frames* (because the data in those frames overrides the data in the key frames).

### The Format of Key Frame Samples

The data in both key frames and override frames is stored in atom containers. (Indeed, atom containers were introduced in QuickTime version 2.1 primarily for the purpose of organizing sprite media data.) A key frame atom container contains a child atom of type `kSpriteAtomType` for each sprite in the key frame. This atom contains leaf atoms that define the initial properties of the sprite. The atom IDs of the sprite atoms are numbered sequentially, starting at 1; these atom IDs are also called *sprite IDs*. **Figure 5** shows the basic structure of a key frame sample.

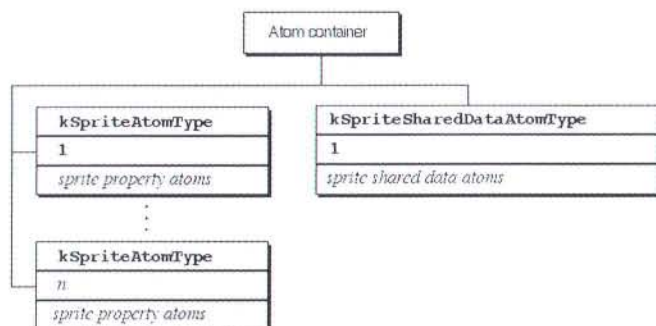


**Figure 5:** The structure of a key frame sample

A sprite atom contains child atoms that define the initial properties of the sprite. It can contain a child for each of the five basic sprite properties, as well as an atom (of type `kSpriteNameAtomType`) that defines the sprite's name. **Figure 6** shows the structure of a sprite atom.

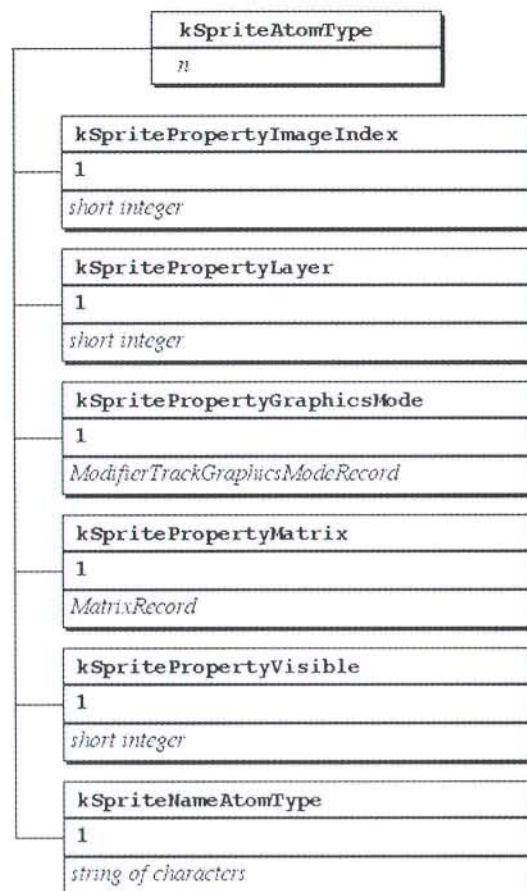


**Figure 6:** The structure of a sprite atom

As **Figure 5** indicates, a key frame atom container also contains a single child atom of type `kSpriteSharedDataAtomType` (with atom ID 1) which contains the image data for all the sprites. This atom contains one *sprite images container atom*, of type `kSpriteImagesContainerAtomType` (also with atom ID 1). This atom, in turn, contains one atom of type `kSpriteImageAtomType` for each individual image in the key frame. Note that all the images for all the sprites are contained in the single images container atom. **Figure 7** shows the structure of a sprite shared data atom.

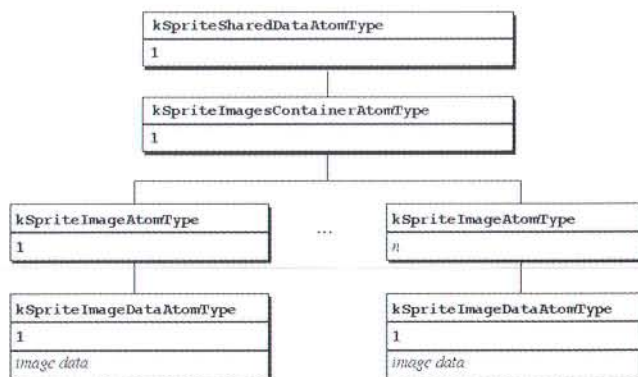
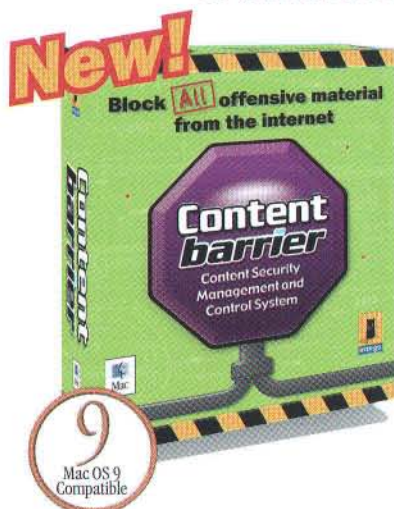


**Figure 7:** The structure of the shared sprite data

# You think your kids are safe on the Net. Think again...



## ContentBarrier. The new Parental Control solution for your Macintosh.



You don't let your kids read just anything, do you? Well, when they surf the Internet, they can see whatever they want, unless you're there to watch over their shoulders. With ContentBarrier, the Internet is a safer place for your children.

ContentBarrier sets up a protective wall around your computer. Its pre-defined categories let you choose what you don't want your children to see, and you can also create your own custom categories. Inappropriate web sites are blocked, shielding your children from things they are too young for.

ContentBarrier blocks adult web sites, sites with subjects not fit for children, and even blocks chats when predatory language is used. It blocks all offensive content coming from the Internet.

ContentBarrier works with multiple users; if you have several children, you can set different criteria, corresponding to their age or maturity. You can choose whether they have access to newsgroups, e-mail, or whether they can download files. You can set the program to let them only use the Internet at certain times, and on specific days. The program can even send you e-mail, automatically, when certain events occur.

ContentBarrier keeps a complete log of all web sites visited, whether blocked or not. This allows you to have a full record of exactly what your children are doing on the Internet. You can also take an inventory of your children's computer, and find all pictures, movies, music files and web pages.

ContentBarrier makes the Internet a safer place for your children.

Licences available  
5, 10, 20+

[www.intego.com](http://www.intego.com)



ContentBarrier is fully compatible with Intego's acclaimed NetBarrier and VirusBarrier

**CDW**  
800-509-4239

New sold at  
**ClubMac**  
www.clubmac.com  
800-258-2622

@**DEPOT**  
800-622-3381

**J&R**  
800-221-8180

**MacMall**  
800-217-9492

**MacNN**  
★★★★★

**MacWorld**  
★★★★★

**APPLE WIZARDS**  
★★★★★

**MacWarehouse**  
800-255-6227

**macZone**  
800-436-0606

**Outpost.com**  
877-688-7678

**COMPUSA**

**MICRO CENTER**  
www.microcenter.com

**buyonet**  
www.buyonet.com

**Macworld**  
★★★★★

**MACGENERATION**



Mac and the Mac logo are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. Outpost.com™ is a trademark of Cyberian Outpost, Inc. All other trademarks belong to their respective owners. Illustration Mona Daly ©2000 Artville Stock Images.

## The Format of Override Samples

The structure of an override sample is somewhat simpler than that of a key frame sample, largely because an override sample does not contain any image data. An override sample is an atom container that contains a sprite atom (of type `kSpriteAtomType`) for each sprite that is being animated by that override sample. The sprite atoms contain child atoms for each of the properties that are changing from the previous key frame or override sample. **Figure 8** shows the structure of a typical override sample.

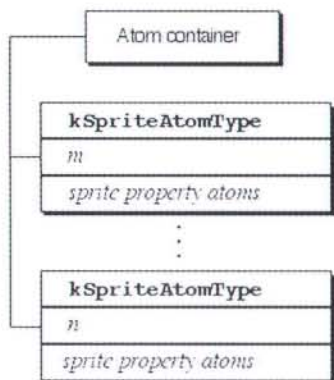


**Figure 8:** The structure of an override sample

The ID of a sprite atom in the override sample should be the same as the ID of the sprite atom in the key frame atom whose data that the override atom is overriding.

### CREATING SPRITE TRACKS

As we've just learned, a sprite track consists of key frame samples that contain the images for the sprites in a track and the initial properties of those sprites, and override samples that change one or more of the properties of those sprites. In both cases, the sample data is contained in an atom container. Building a sprite track is therefore largely a matter of creating the appropriate atom containers and inserting them at the desired times in the sprite track media.

### Creating Sprite Tracks and Media

When the user selects an item in the Test menu, QTSprites calls the `QApp_HandleMenu` function, which is shown in Listing 1.

#### Listing 1: Handling items in the Test menu

```

Boolean QApp_HandleMenu (UInt16 theMenuItem)
{
    Boolean    myIsHandled = false;

    switch (theMenuItem) {

        case IDM_MAKE_ICONS_MOVIE:
        case IDM_MAKE_PENGUIN_MOVIE:
        case IDM_MAKE_SPACE_MOVIE:
            QTSprites_CreateSpritesMovie(theMenuItem);
            myIsHandled = true;
            break;
    }
}

```

```

case IDM_USE_BACKGROUND_IMAGE:
    gUseBackgroundPicture = !gUseBackgroundPicture;
    myIsHandled = true;
    break;

} // switch (theMenuItem)

return(myIsHandled);
}

```

As you can see, we call the function `QTSprites_CreateSpritesMovie` to create each of the three sample movies, passing in the menu item so that we know which movie to create. `QTSprites_CreateSpritesMovie` is defined in Listing 2.

#### Listing 2: Creating a sprite movie

```

QTSprites_CreateSpritesMovie
OSErr QTSprites_CreateSpritesMovie (UInt16 theMenuItem)
{
    Movie        myMovie = NULL;
    Track        myTrack = NULL;
    Media        myMedia = NULL;
    FSSpec       myFile;
    Boolean      myIsSelected = false;
    Boolean      myIsReplacing = false;
    Fixed        myHeight = 0;
    Fixed        myWidth = 0;
    StringPtr    myPrompt =
        QTUtils_ConvertCToPascalString(kSpriteSavePrompt);
    StringPtr    myFileName =
        QTUtils_ConvertCToPascalString(kSpriteSaveMovieFileName);
    long         myFlags = createMovieFileDeleteCurFile |
        createMovieFileDontCreateResFile;
    short        myResRefNum = 0;
    short        myResID = movieInDataForkResID;
    OSErr        myErr;

    // prompt the user for the destination file name
    QTFrame_PutFile(myPrompt, myFileName, &myFile,
        &myIsSelected, &myIsReplacing);
    myErr = myIsSelected ? noErr : userCanceledErr;
    if (!myIsSelected)
        goto bail;

    // create a movie file for the destination movie
    myErr = CreateMovieFile(&myFile, FOUR_CHAR_CODE('TVOD'),
        smSystemScript, myFlags, &myResRefNum, &myMovie);
    if (myErr != noErr)
        goto bail;

    // create the sprite track and media
    QTSprites_GetMovieSize(theMenuItem, &myHeight, &myWidth);

    myTrack = NewMovieTrack(myMovie, myWidth, myHeight,
        kNoVolume);
    myMedia = NewTrackMedia(myTrack, SpriteMediaType,
        kSpriteMediaTimeScale, NULL, 0);

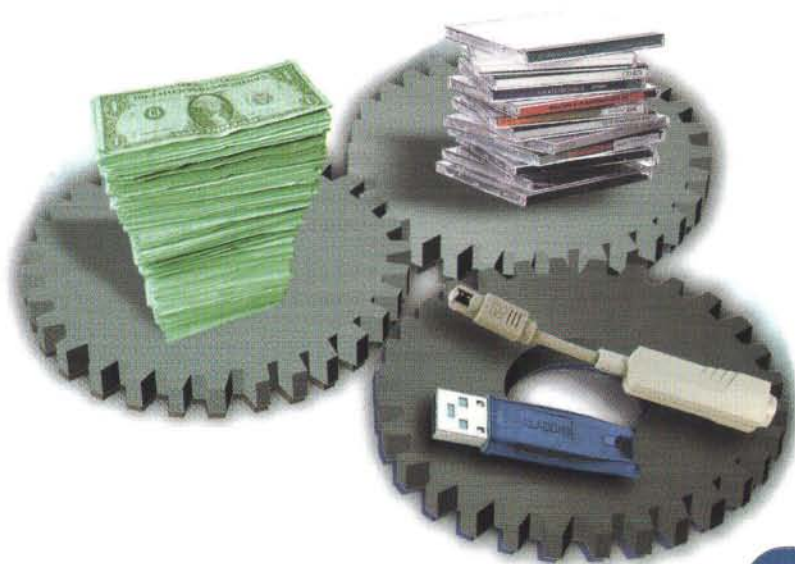
    BeginMediaEdits(myMedia);

    // add the appropriate samples to the sprite media
    switch (theMenuItem) {
        case IDM_MAKE_ICONS_MOVIE:
            QTSprites_AddIconMovieSamplesToMedia(myMedia);
            break;
        case IDM_MAKE_PENGUIN_MOVIE:
            QTSprites_AddPenguinMovieSamplesToMedia(myMedia);
            break;
        case IDM_MAKE_SPACE_MOVIE:
            QTSprites_AddSpaceMovieSamplesToMedia(myMedia);
            break;
        default:
            goto bail;
    }
}

```

# Software Developers

## HASP4 for Mac - Turn Your Lost Software Into Profits!



Turn your lost software into profits. By preventing unauthorized software usage and duplication, HASP4 for Mac protects intellectual property rights and increases revenues. It enforces the legitimate use and licensing of software applications. HASP4 for Mac controls software distribution and expands sales channels. Companies such as 3M, Quark, Lucent, HP, Cadlink and Scitex know that they can count on HASP to protect their software products.

**Order your HASP4 for Mac Developer's Kit at:**

**[www.eAladdin.com/mactech](http://www.eAladdin.com/mactech)**

**Or call 800-223-4277 EST or  
800-562-2543 WST, CST & MST.**

**ALADDIN**

Securing the Global Village

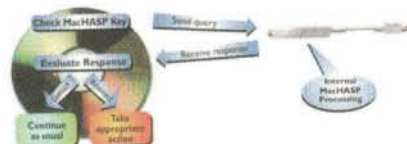
**NOW  
SUPPORTS  
OS X!**

### Identification Card

**HASP4**  
The Professional Software  
Protection System

|                                |                                                                                                                                                           |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Name:</b>                   | HASP4 for Mac                                                                                                                                             |
| <b>Main Mission:</b>           | To protect your software & Increase your revenues                                                                                                         |
| <b>HASP is:</b>                | A small hardware key that connects to the computer and prevents unauthorized use of your software.                                                        |
| <b>Features &amp; benefits</b> | Prevents software piracy<br>Controls software distribution<br>Protects intellectual property<br>Increases revenues<br>Reduces software distribution costs |

#### Product Diagram:



International [HASP@eAladdin.com](mailto:HASP@eAladdin.com) North America [HASPus@eAladdin.com](mailto:HASPus@eAladdin.com) UK [HASPuk@eAladdin.com](mailto:HASPuk@eAladdin.com) Germany [HASPde@eAladdin.com](mailto:HASPde@eAladdin.com)

Benelux [HASPnl@eAladdin.com](mailto:HASPnl@eAladdin.com) France [HASPfr@eAladdin.com](mailto:HASPfr@eAladdin.com) Israel [HASPil@eAladdin.com](mailto:HASPil@eAladdin.com) South Africa [HASPza@eAladdin.com](mailto:HASPza@eAladdin.com)

Brazil [HASPbr@eAladdin.com](mailto:HASPbr@eAladdin.com) Japan [HASPjp@eAladdin.com](mailto:HASPjp@eAladdin.com) Russia [HASPru@eAladdin.com](mailto:HASPru@eAladdin.com) Spain [HASPes@eAladdin.com](mailto:HASPes@eAladdin.com) China [HASPcn@eAladdin.com](mailto:HASPcn@eAladdin.com)

```

EndMediaEdits(myMedia);

// add the media to the track
InsertMediaIntoTrack(myTrack, 0, 0,
    GetMediaDuration(myMedia), fixed!);

// set the sprite track properties
QTSprites_SetTrackProperties(myMedia, theMenuItem);

// add the movie resource to the movie file
myErr = AddMovieResource(myMovie, myResRefNum, &myResID,
    myFile.name);

bail:
if (myResRefNum != 0)
    CloseMovieFile(myResRefNum);

if (myMovie != NULL)
    DisposeMovie(myMovie);

free(myPrompt);
free(myFileName);

return(myErr);
}

```

The QTSprites\_CreateSpritesMovie function is remarkably similar to each of the other movie-creation functions we've used earlier in this series of articles (compare, for instance, the QTMM\_CreateVideoMovie function in "Making Movies" in *MacTech*, June 2000). There are only three main additions for QTSprites. First, since we want to be able to create any one of three different sprite movies, we call the function QTSprites\_GetMovieSize to get the desired size for each of those movies. QTSprites\_GetMovieSize is defined in Listing 3.

### Listing 3: Getting the size of a sprite movie

```

QTSprites_GetMovieSize
void QTSprites_GetMovieSize (UInt16 theMenuItem,
    Fixed *theHeight, Fixed *theWidth)
{
    if ((theHeight == NULL) || (theWidth == NULL))
        return;

    switch (theMenuItem) {
        case IDM_MAKE_ICONS_MOVIE:
            *theWidth = (long)kIconSpriteTrackWidth << 16;
            *theHeight = (long)kIconSpriteTrackHeight << 16;
            break;
        case IDM_MAKE_PENGUIN_MOVIE:
            *theWidth = (long)kPenguinSpriteTrackWidth << 16;
            *theHeight = (long)kPenguinSpriteTrackHeight << 16;
            break;
        case IDM_MAKE_SPACE_MOVIE:
            *theWidth = (long)kSpaceSpriteTrackWidth << 16;
            *theHeight = (long)kSpaceSpriteTrackHeight << 16;
            break;
    }
}

```

This is pretty simple stuff; we just convert some long integer constants to the Fixed data type and return them to the caller.

The second difference between QTSprites\_CreateSpritesMovie and our earlier movie-creation functions is that we use the menu item number passed in to select the appropriate function for adding samples to the sprite media. And, third, once we've added those samples to the media, we call the QTSprites\_SetTrackProperties function to set some sprite track properties. We'll consider sprite track properties in more detail.



## SUPERIOR Products for Your POWERBOOK

### Xcarét Pro Expansion Bay CDRW Drives

- Bootable! • Hot Swappable • Fast Backup or File Transfer
- Toast™ 4 Software • Protective MCE Carrying Case
- Available for PowerBook G3 98 (Wall Street) or PowerBook G3 99 & 2000 (Lombard & Pismo)

**iTunes and Disc Burner Compatible!**



### Internal Hard Drive Upgrades for PowerBooks



- The BEST Internal Hard Drive Upgrade Kits for your PowerBook!
- Package includes hard drives from the same manufacturers that shipped in the PowerBook from the factory, bundled with everything you need for a successful installation
- High-Speed 10GB, 20GB, and 30GB Capacities

### DataShuttle Xtreme Portable Hard Drive

- The Most VERSATILE Portable Hard Drives Anywhere
- FireWire Hard Drives with Flexible USB and PC Card Options
- High-Speed 10GB, 20GB, and 30GB Capacities
- Also available, the Do-It-Yourself DataShuttle Xtreme Kit - Convert Your old Internal Drive into a Portable External Drive!



### Xcarét Pro Expansion Bay Hard Drives

- Ultra Fast! • Hot Swappable • Bootable
- Compact • Protective MCE Carrying Case
- High-Speed 10GB, 20GB, and 30GB Capacities
- Also available, the Xcarét Pro Expansion Bay Kit - Convert Your old Drive into an Expansion Bay Drive!
- Available for PowerBook G3 98 or PowerBook G3 99 & 2000



Great for Mac OS X Testing

### USB FlexLight

"Illuminate Your Workspace"



- Conveniently lights up your keyboard and surrounding paperwork. • Twists and Bends so You can Direct the Light Where You Want it • Powered by the USB Port
- Comes with Both a Clear and a Red Diffuser - You Choose the Type of Light to Suit Your Situation • Very Low Power Draw



Mac Components Engineered



Distributed by:  
INGRAM MICRO

For more info:  
**www.mctech.com**  
800.500.0622  
949.458.0880  
Contact us for a dealer near you.

## Setting Sprite Properties

A key frame sample is an atom container that contains an atom of type `kSpriteAtomType` for each sprite in the frame (which contains the initial properties of the sprite) and an atom of type `kSpriteSharedDataAtomType` (which contains atoms that hold the sprite images). So the first thing we need to do is create an atom container, like this:

```
myErr = QTNewAtomContainer(&mySample);
```

Let's begin by adding the sprite atoms to the key frame sample. A sprite atom is itself an atom container, because it contains child atoms for each of the initial sprite properties we want to assign it. (Any properties we don't explicitly define in a key frame sample are set to default values.) So we need to create another atom container, like so:

```
myErr = QTNewAtomContainer(&mySpriteData);
```

Now we want to add one or more property atoms to the sprite atom. For the icon movie, we want to set the initial location, visibility state, layer, and image index. We can set the image index, for instance, like this:

```
short    myIndex = 1;

myIndex = EndianS16_NtoB(myIndex);
myErr = QTInsertChild(mySpriteData, kParentAtomIsContainer,
    kSpritePropertyImageIndex, 1, 0, sizeof(short),
    &myIndex, NULL);
```

This code inserts a child of type `kSpritePropertyImageIndex` into the sprite atom, making sure that the atom data (in this case, a short integer whose value is 1) is in big-endian format.

Similarly, we can set the initial visibility state of the icon sprite using these lines of code:

```
short    isVisible = true;

isVisible = EndianS16_NtoB(isVisible);
myErr = QTInsertChild(mySpriteData, kParentAtomIsContainer,
    kSpritePropertyVisible, 1, 0, sizeof(short),
    &isVisible, NULL);
```

(You might have thought that the default value for the visibility state of a sprite would be `true`, but sadly that's not so. So we need to explicitly configure our sprites to be visible or they won't be drawn.)

To increase the readability of our code, we'll define a utility function called `SpriteUtils_SetSpriteData` that allows us to set the main sprite properties in one fell swoop. Then we can define the initial state of the icon sprite like this:

```
myLocation.h = 32;
myLocation.v = 32;
isVisible = true;
myLayer = -1;
myIndex = 1;

SpriteUtils_SetSpriteData(mySpriteData, &myLocation,
    &isVisible, &myLayer, &myIndex, NULL, NULL, NULL);
```

The `SpriteUtils_SetSpriteData` function is defined in the file `SpriteUtilities.c`; its definition is shown in Listing 4.



Copyright 2001, Terra Soft Solutions, Inc.

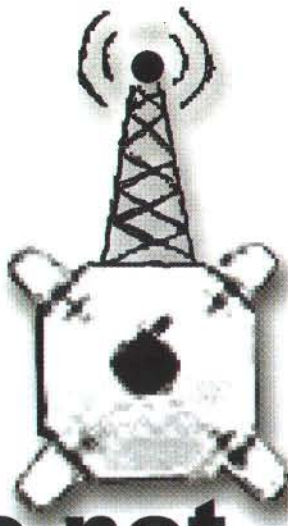
# 2.0

## YDL 2.0 for PowerPC

by Terra Soft Solutions, Inc.

*Featuring kernel 2.4, KDE 2, XFree86 4  
and a graphical installer like none other.*

**www.yellowdoglinux.com**  
*Redefining your PowerPC experience.*



**MacOS  
Radio.net**  
**www.macosradio.net**  
(a Mardun Software Ltd. company)

•Talk radio about the Mac,  
only on the Internet.

•No more surfing to 100  
different sites to get your  
MacOS news.

•Interesting guests from  
companies you care about  
and sources you listen to.

**ALL MAC  
ALL THE TIME.**

**For more information about Internet  
radio, QuickTime Streaming Servers  
and how you can develop the power  
of MacOS X Server for media  
streaming today, e-mail  
sales@macosradio.net**

**or call  
573-443-0301**

**Mardun**

©2001 Mardun Software Ltd.

#### Listing 4: Setting properties of a sprite

SpriteUtils\_SetSpriteData

```
OSErr SpriteUtils_SetSpriteData (
    QTAAtomContainer theSprite,
    Point *theLocation,
    short *theVisible,
    short *theLayer,
    short *theImageIndex,
    ModifierTrackGraphicsModeRecord *theGraphicsMode,
    StringPtr theSpriteName,
    QTAAtomContainer theActionAtoms)
{
    QTAAtom        myPropertyAtom;
    OSErr          myErr = noErr;

    // set the sprite location data
    if (theLocation != NULL) {
        MatrixRecord myMatrix;

        SetIdentityMatrix(&myMatrix);
        myMatrix.matrix[2][0] = ((long)theLocation->h << 16);
        myMatrix.matrix[2][1] = ((long)theLocation->v << 16);
        EndianUtils_MatrixRecord_NtoB(&myMatrix);

        myPropertyAtom = QTFindChildByIndex(theSprite,
            kParentAtomIsContainer,
            kSpritePropertyMatrix, 1, NULL);
        if (myPropertyAtom == 0)
            myErr = QTInsertChild(theSprite,
                kParentAtomIsContainer,
                kSpritePropertyMatrix, 1, 0,
                sizeof(MatrixRecord), &myMatrix, NULL);
        else
            myErr = QTSetAtomData(theSprite, myPropertyAtom,
                sizeof(MatrixRecord), &myMatrix);

        if (myErr != noErr)
            goto bail;
    }

    // set the sprite visibility state
    if (theVisible != NULL) {
        short myVisible = *theVisible;

        myVisible = EndianS16_NtoB(myVisible);

        myPropertyAtom = QTFindChildByIndex(theSprite,
            kParentAtomIsContainer,
            kSpritePropertyVisible, 1, NULL);
        if (myPropertyAtom == 0)
            myErr = QTInsertChild(theSprite,
                kParentAtomIsContainer,
                kSpritePropertyVisible, 1, 0,
                sizeof(short), &myVisible, NULL);
        else
            myErr = QTSetAtomData(theSprite, myPropertyAtom,
                sizeof(short), &myVisible);

        if (myErr != noErr)
            goto bail;
    }

    // set the sprite layer
    if (theLayer != NULL) {
        short myLayer = *theLayer;

        myLayer = EndianS16_NtoB(myLayer);

        myPropertyAtom = QTFindChildByIndex(theSprite,
            kParentAtomIsContainer,
            kSpritePropertyLayer, 1, NULL);
        if (myPropertyAtom == 0)
            myErr = QTInsertChild(theSprite,
                kParentAtomIsContainer,
                kSpritePropertyLayer, 1, 0,
                sizeof(short), &myLayer, NULL);
        else
            myErr = QTSetAtomData(theSprite, myPropertyAtom,
                sizeof(short), &myLayer);

        if (myErr != noErr)
            goto bail;
    }
}
```

DESIGNED FOR THE WAY YOU WORK

*I'd rather create clocks than invoices.*

***If I wanted to keep books all day, I'd have been an accountant.***



mind your own business

**MYOB**

Roy Fox, Founder, Fox Clocks, Hollis, Maine  
One of 450,000 MYOB software users

**MYOB**

**Small Business.  
Smart Solutions.™**

800.322.MYOB  
[myob.com/us](http://myob.com/us)

MYOB software is the simplest, most powerful, most complete solution for managing my company on the Mac, from the day to day to the bottom line.

**Antique frames. Quartz movements. That's my business.**

**MYOB software works for me.**

```

}

// set the sprite image index
if (theImageIndex != NULL) {
    short    myImageIndex = *theImageIndex;

    myImageIndex = EndianS16_NtoB(myImageIndex);

    myPropertyAtom = QTFindChildByIndex(theSprite,
        kParentAtomIsContainer,
        kSpritePropertyImageIndex, 1, NULL);
    if (myPropertyAtom == 0)
        myErr = QTInsertChild(theSprite,
            kParentAtomIsContainer,
            kSpritePropertyImageIndex, 1, 0,
            sizeof(short), &myImageIndex, NULL);
    else
        myErr = QTSetAtomData(theSprite, myPropertyAtom,
            sizeof(short), &myImageIndex);

    if (myErr != noErr)
        goto bail;
}

// set the sprite graphics mode
if (theGraphicsMode != NULL) {
    ModifierTrackGraphicsModeRecord    myGraphicsMode;

    myGraphicsMode.graphicsMode =
        EndianU32_NtoB(theGraphicsMode->graphicsMode);
    myGraphicsMode.opColor.red =
        EndianU16_NtoB(theGraphicsMode->opColor.red);
    myGraphicsMode.opColor.green =
        EndianU16_NtoB(theGraphicsMode->opColor.green);
    myGraphicsMode.opColor.blue =
        EndianU16_NtoB(theGraphicsMode->opColor.blue);

    myPropertyAtom = QTFindChildByIndex(theSprite,
        kParentAtomIsContainer,
        kSpritePropertyGraphicsMode, 1, NULL);
    if (myPropertyAtom == 0)
        myErr = QTInsertChild(theSprite,
            kParentAtomIsContainer,
            kSpritePropertyGraphicsMode, 1, 0,
            sizeof(myGraphicsMode), &myGraphicsMode,
            NULL);
    else
        myErr = QTSetAtomData(theSprite, myPropertyAtom,
            sizeof(myGraphicsMode), &myGraphicsMode);

    if (myErr != noErr)
        goto bail;
}

// set the sprite name
if (theSpriteName != NULL) {
    QTAtom    mySpriteNameAtom;

    mySpriteNameAtom = QTFindChildByIndex(theSprite,
        kParentAtomIsContainer,
        kSpriteNameAtomType, 1, NULL);
    if (mySpriteNameAtom == 0)
        myErr = QTInsertChild(theSprite,
            kParentAtomIsContainer,
            kSpriteNameAtomType, 1, 0,
            theSpriteName[0] + 1, theSpriteName,
            NULL);
    else
        myErr = QTSetAtomData(theSprite, mySpriteNameAtom,
            theSpriteName[0] + 1, theSpriteName);

    if (myErr != noErr)
        goto bail;
}

// set the action atoms
if (theActionAtoms != NULL)
    myErr = QTInsertChildren(theSprite,
        kParentAtomIsContainer,
theActionAtoms);

bail:

```

```

    if ((myErr != noErr) && (theSprite != NULL))
        QTRemoveChildren(theSprite, 0);

    return(myErr);
}

```

For each parameter that is not `NULL`, `SpriteUtils_SetSpriteData` looks to see whether the sprite atom container already contains an atom of that the corresponding type (by calling `QTFindChildByIndex`). If it does contain such an atom, then `SpriteUtils_SetSpriteData` calls `QTSetAtomData` to reset the data in that atom; otherwise, it calls `QTInsertChild` to add an atom of that type. In all cases, the data passed in is converted to big-endian format before being inserted into an atom.

Now that the `mySpriteData` atom container holds a child atom for each initial property we want to set, we need to add it to the key frame sample atom container, `mySample`. Once again, we'll define a utility function to help us out:

```

SpriteUtils_AddSpriteToSample(mySample, mySpriteData,
kQTIconSpriteAtomID);

```

`SpriteUtils_AddSpriteToSample` is defined in Listing 5.

#### Listing 5: Adding a sprite data atom to a sample container

```

OSErr SpriteUtils_AddSpriteToSample
    (QTAtomContainer theSample, QTAtomContainer theSprite,
     QTAtomID theSpriteID)
{
    QTAtom    mySpriteAtom = 0;
    OSErr     myErr = paramErr;

    // see if the sample already contains a sprite atom of the specified ID
    mySpriteAtom = QTFindChildByID(theSample,
        kParentAtomIsContainer,
        kSpriteAtomType, theSpriteID, NULL);
    if (mySpriteAtom != 0)
        goto bail;

    // here, the index 0 means to append the sprite to the sample
    myErr = QTInsertChild(theSample, kParentAtomIsContainer,
        kSpriteAtomType, theSpriteID, 0, 0, NULL,
        &mySpriteAtom);
    if (myErr != noErr)
        goto bail;

    myErr = QTInsertChildren(theSample, mySpriteAtom,
        theSprite);

    bail:
    return(myErr);
}

```

As you can see, `SpriteUtils_AddSpriteToSample` first calls `QTFindChildByID` to determine whether the specified atom container already contains a sprite atom with the specified sprite ID. If there is an atom of that ID already in the sample, `SpriteUtils_AddSpriteToSample` returns an error to the caller. Otherwise, it calls `QTInsertChild` to create a sprite atom of that ID in the sample atom container. Then it calls `QTInsertChildren` to copy the children from the sprite atom container passed as a parameter into the newly inserted atom in the sample.

#### Setting Sprite Images

So far, then, we've created a sprite atom that contains the desired initial properties of the icon sprite and added it to the



# OPENBASE SQL

# 6.5

## DATABASE ENGINE

**OpenBase SQL 6.5**  
*The high-performance Database for MacOS X*

- OPENBASE FEATURES**
- JDBC Driver
  - WebObjects Adaptors
  - REALbasic Plugin
  - Multi-Threaded Server
  - Row Level Locking
  - 100 MB Searchable Blobs
  - Database Replication
  - Database Management Tools
  - Graphical Schema Design Tools
  - Remote Administration

*"We process over 2 million transactions each day. OpenBase performance has been outstanding."*  
**FlightArrivals.com**

### AMAZING SPEED      AMAZING POWER

### [ A M A Z I N G L Y   A F F O R D A B L E ]

With over **2 Million Transactions** per day, **FlightArrivals.com** flies with **OpenBase SQL**

**Take OpenBase SQL for a Test Flight Today!**  
**Get Your FREE Developer's License at [www.openbase.com](http://www.openbase.com)**



key frame atom container `mySample`. Now we need to create an atom of type `kSpriteSharedDataAtomType` and add the required subatoms to it that contain the sprite image data. In the case of the icon sprite movie, we need to add two images to that atom, one for the old QuickTime extension icon and one for the new icon. The function `QTSprites_AddIconMovieSamplesToMedia` does this by executing these lines of code:

```
myKeyColor.red = myKeyColor.green = myKeyColor.blue = 0xffff;

SpriteUtils_AddPictImageToKeyFrameSample(mySample,
    kOldQTIconID, &myKeyColor, 1, NULL, NULL);
SpriteUtils_AddPictImageToKeyFrameSample(mySample,
    kNewQTIconID, &myKeyColor, 2, NULL, NULL);
```

Once again, we're relying on a function defined in the file `SpriteUtilities.c` to hide the nitty-gritty details of building the required atom from our main application. `SpriteUtils_AddPictImageToKeyFrameSample` (defined in Listing 6) reads a picture from our application's resource fork, recompresses the picture with the specified color as a transparency color, and then adds it to the key frame sample atom container.

**Listing 6: Adding compressed image data to a key frame sample**

```
SpriteUtils_AddPictImageToKeyFrameSample
OSErr SpriteUtils_AddPictImageToKeyFrameSample
(QTAtomContainer theKeySample, short thePictID,
 RGBColor *theKeyColor, QTAtomID theID,
 FixedPoint *theRegistrationPoint,
 StringPtr theImageName)
```

```
{
    PicHandle      myPicture = NULL;
    Handle         myCompressedPicture = NULL;
    ImageDescriptionHandle myImageDesc = NULL;
    OSErr          myErr = noErr;

    // get picture from resource
    myPicture = (PicHandle)GetPicture(thePictID);
    if (myPicture == NULL)
        myErr = resNotFound;

    if (myErr != noErr)
        goto bail;

    DetachResource((Handle)myPicture);

    // convert it to image data compressed by the animation compressor
    myErr = ICUtils_RecompressPictureWithTransparency
        (myPicture, theKeyColor, NULL, &myImageDesc,
         &myCompressedPicture);
    if (myErr != noErr)
        goto bail;

    // add it to the key sample
    HLock(myCompressedPicture);
    myErr = SpriteUtils_AddCompressedImageToKeyFrameSample
        (theKeySample, myImageDesc,
         GetHandleSize(myCompressedPicture),
         *myCompressedPicture, theID, theRegistrationPoint,
         theImageName);

bail:
    if (myPicture != NULL)
        KillPicture(myPicture);

    if (myCompressedPicture != NULL)
        DisposeHandle(myCompressedPicture);

    if (myImageDesc != NULL)
        DisposeHandle((Handle)myImageDesc);

    return(myErr);
}
```

`SpriteUtils_AddPictImageToKeyFrameSample` does most of its work by calling two other utility functions, `ICUtils_RecompressPictureWithTransparency` and `SpriteUtils_AddCompressedImageToKeyFrameSample`. We won't dissect either of these functions in detail here, as that would take us too far afield. `SpriteUtils_AddCompressedImageToKeyFrameSample` really just does what you'd imagine to build the atom container illustrated in **Figure 7**.

### Adding the Key Frame Sample to the Sprite Media

So, we've managed to build the key frame sample. Now we just need to add it to the sprite media. To do this, we first need to create a handle to a sprite sample description, which we'll pass to `AddMediaSample`. A *sprite sample description* is defined by the `SpriteDescription` data type, declared like this:

```
struct SpriteDescription {
    long        descSize;
    long        dataFormat;
    long        resvd1;
    short       resvd2;
    short       dataRefIndex;
    long        version;
    OSType      decompressorType;
    long        sampleFlags;
};
```

We allocate a handle to a sprite sample description by calling `NewHandleClear`:

```
mySampleDesc = (SampleDescriptionHandle)
NewHandleClear(sizeof(SpriteDescription));
```

The `decompressorType` field of the sprite sample description specifies the type of compressor used to compress the sample data (or 0 if no compression is used). The sprite media handler supports compressed sample data; the only restriction is that the compressor used to compress the data must be lossless. (A compressor is *lossless* if the result of compressing some data and then decompressing it yields the original data unchanged; otherwise, the compressor is *lossy*.) For the moment, we'll just use the uncompressed data that we've created. So we can just go right ahead and call `AddMediaSample`:

```
myErr = AddMediaSample(theMedia, (Handle)mySample, 0,
    GetHandleSize(mySample), kSpriteMediaFrameDurationIcon,
    mySampleDesc, 1, 0, NULL);
```

We can use yet another utility function defined in `SpriteUtilities.c`:

```
SpriteUtils_AddSpriteSampleToMedia(theMedia, mySample,
    kSpriteMediaFrameDurationIcon, true, NULL);
```

Listing 7 shows our definition of `SpriteUtils_AddSpriteSampleToMedia`.

### Listing 7: Adding a sprite sample to the sprite media

```
SpriteUtils_AddSpriteSampleToMedia
OSErr SpriteUtils_AddSpriteSampleToMedia (Media theMedia,
    QTAAtomContainer theSample, TimeValue theDuration,
    Boolean isKeyFrame, TimeValue *theSampleTime)
{
    SampleDescriptionHandle mySampleDesc = NULL;
    OSErr                    myErr = noErr;
```

**“Listen Different”**

[www.macshowlive.com](http://www.macshowlive.com)

Every week, we're streaming live, interactive, and talking...

**100% Mac!**

Shawn King

Hilary Percy

• News • Basics • Gaming • Mac Pro • Tech Tips • Feature Interviews

listen anytime @ [macshowlive.com](http://macshowlive.com)

**MAC LIVE SHOW**



Marketing Rule no.1

**MAKE YOUR PRODUCT EASIER TO BUY, YOU'LL SELL MORE OF IT.**

You may have developed the best software in the world, but if nobody knows about it, or can find it, you won't sell much of it. With eSellerate, your application literally sells itself. Providing your customers with the power of instant gratification. And providing you with an opportunity of nearly limitless potential. [www.esellerate.net](http://www.esellerate.net)



**eSellerate**  
The new way to sell software.

From MindVision Software, creator of



```

mySampleDesc = (SampleDescriptionHandle)
    NewHandleClear(sizeof(SpriteDescription));
if (mySampleDesc == NULL) {
    myErr = MemError();
    goto bail;
}

myErr = AddMediaSample(theMedia,
    (Handle)theSample,
    0,
    GetHandleSize(theSample),
    theDuration,
    mySampleDesc,
    1,
    (short)(isKeyFrame ? 0 : mediaSampleNotSync),
    theSampleTime);

bail:
if (mySampleDesc != NULL)
    DisposeHandle((Handle) mySampleDesc);
return(myErr);
}

```

Note that `SpriteUtils_AddSpriteSampleToMedia` adds the specified sample data as a key frame sample or an override frame sample, depending on the value of the `isKeyFrame` parameter.

### Creating Override Samples

With these sprite utility functions at our disposal, it's now quite easy to create some override samples and add them to the sprite media. For the icon movie, we want to create 99 override samples. Each override sample will move the icon 2 pixels to the right; furthermore, when the icon reaches the halfway point, we'll change the image index from 1 to 2. Listing 8 shows the code we use to add those override samples to the sprite media.

#### Listing 8: Adding some override samples

```

QTSprites_AddIconMovieSamplesToMedia
for (myCount = 1; myCount <= kNumOverrideSamples; myCount++)
{
    QTRemoveChildren(mySample, kParentAtomIsContainer);
    QTRemoveChildren(mySpriteData, kParentAtomIsContainer);

    // every frame, bump the icon's location
    myLocation.h += 2;

    // change icon half way thru
    if (myCount == kNumOverrideSamples / 2)
        myIndex = 2;

    SpriteUtils_SetSpriteData(mySpriteData, &myLocation, NULL,
        NULL, &myIndex, NULL, NULL, NULL);
    SpriteUtils_AddSpriteToSample(mySample, mySpriteData,
        kQTIconSpriteAtomID);
    SpriteUtils_AddSpriteSampleToMedia(theMedia, mySample,
        kSpriteMediaFrameDurationIcon, false, NULL);
}

```

### Setting Sprite Track Properties

We saw earlier that we can use a sprite image as the background of a sprite track by setting the layer of the image to `kBackgroundSpriteLayerNum`. But what if we want to have a solid background color for the entire sprite track? We could of course create a sprite image of the proper size and color and then put it into the first key frame sample of the sprite track. The sprite media handler, however, provides a better method to set a solid background color by allowing us to set the sprite track's

background color property. The background color property is one of several *sprite track properties* that control global aspects of the sprite track. The currently defined sprite track properties are accessed using these constants:

```

enum {
    kSpriteTrackPropertyBackgroundColor      = 101,
    kSpriteTrackPropertyOffscreenBitDepth    = 102,
    kSpriteTrackPropertySampleFormat         = 103,
    kSpriteTrackPropertyScaleSpritesToScaleWorld = 104,
    kSpriteTrackPropertyHasActions           = 105,
    kSpriteTrackPropertyVisible              = 106,
    kSpriteTrackPropertyQTIdleEventsFrequency = 107
};

```

To set one or more sprite track properties, we need to create a *media property atom*, an atom container that contains a child atom for each of the properties we want to set. The atom type of the child atom should be set to one of these sprite property constants, and the atom ID should be set to 1. The type of the atom data varies from property to property. For the background color sprite track property, the atom data is an `RGBColor` structure. Once we've constructed the media property atom, we attach it to the sprite track by calling the `SetMediaPropertyAtom` function. Listing 9 shows how we set a solid white background for the penguin sprite movie. (If we don't specify a background sprite image and we don't set the background color property, then we'll get the default sprite track background color, which is black.)

#### Listing 9: Setting the background color of a sprite track

```

QTSprites_SetTrackProperties
void QTSprites_SetTrackProperties (Media theMedia,
    UInt16 theMenuItem)
{
    QTAtomContainer    myTrackProperties;
    RGBColor           myBackgroundColor;
    OSErr              myErr = noErr;

    if (!gUseBackgroundPicture) {
        // add a background color to the sprite track
        QTSprites_GetBackgroundColor(theMenuItem,
            &myBackgroundColor);

        myErr = QTNewAtomContainer(&myTrackProperties);
        if (myErr == noErr) {
            QTInsertChild(myTrackProperties, 0,
                kSpriteTrackPropertyBackgroundColor, 1, 1,
                sizeof(RGBColor), &myBackgroundColor,
                NULL);

            SetMediaPropertyAtom(theMedia, myTrackProperties);

            QTDisposeAtomContainer(myTrackProperties);
        }
    }
}

```

The `kSpriteTrackPropertyOffscreenBitDepth` property specifies the desired bit depth of the offscreen graphics world where the sprite data is drawn before it is copied to the screen. The atom data is of type `short`, and the default value is 0 (which means to use the bit depth of the deepest monitor that intersects the onscreen sprite window). Setting this property can save memory if your sprite graphics are drawn at a lower bit depth than the user's monitor.

The `kSpriteTrackPropertySampleFormat` property specifies the

override sample interpretation mode, which indicates how the sprite media handler interprets override samples. Currently there are two possible modes:

```
enum {
    kKeyFrameAndSingleOverride    = 1L << 1,
    kKeyFrameAndAllOverrides      = 1L << 2
};
```

If the override sample interpretation mode is `kKeyFrameAndSingleOverride`, then the sprite media handler generates the sprite data for a particular override sample by applying the changes in that override sample directly to the key frame sample, ignoring any previous override samples. (This is the default mode.) On the other hand, if the mode is `kKeyFrameAndAllOverrides`, then the sprite media handler generates sprite data by applying the changes in a particular override sample to the data generated by applying the changes in *all* previous override samples to the key frame sample data. In other words, `kKeyFrameAndSingleOverride` specifies that a particular override frame contains absolute changes to key frame data, while `kKeyFrameAndAllOverrides` specifies relative changes.

The `kSpriteTrackPropertyScaleSpritesToScaleWorld` property specifies whether the sprites in the sprite track are rescaled whenever the sprite track is resized. This is useful mostly when the sprite images have been compressed using a resolution-independent codec (for example, the Curve codec). The atom data is of type `Boolean`, and the default value is `false`.

The `kSpriteTrackPropertyVisible` property specifies whether the sprite track is visible. The atom data is of type `Boolean`, and the default value is `true`. It might occasionally be useful to set the sprite track to be invisible if there are other tracks in the movie and you want to allow the user to click on items in those tracks (by putting an invisible sprite track in front that intercepts those clicks).

The remaining two sprite track properties apply only to sprite tracks that contain wired actions. The `kSpriteTrackPropertyHasActions` property specifies whether the sprite track contains any wired actions; the atom data is of type `Boolean` and the default value is `false`. The `kSpriteTrackPropertyQTIdleEventsFrequency` property indicates the desired frequency at which the sprite media handler should send idle events (events of type `kQTIdleEvent`) to the sprite track. In this case, the atom data is of type `UInt32` and the default value is `kNoQTIdleEvents` (which means not to issue any idle events). We'll consider these two sprite track properties in more detail in our article on wired sprites.

### Putting it All Together

Let's see what the two functions `QTSprites_AddIconMovieSamplesToMedia` and `QTSprites_AddPenguinMovieSamplesToMedia` look like in their entirety. Listing 10 shows the definition of `QTSprites_AddIconMovieSamplesToMedia`.



## Win a free iMac®!

Simply log on to:  
[www.powermax.com/mactech.html](http://www.powermax.com/mactech.html)  
and register to win an iMac! It's that simple.

### PowerMax is a great source for OS X!

- Aqua user interface
- Gentle migration
- Super-modern core operating system
- Killer graphics
- Internet savvy

### OS X Survival Bundles: huge internal hard drives and extra RAM!

#### Package 1:

- Internal 75 Gb IBM DeskStar 7200 rpm drive w/5 yr warranty
- 256 Mb Additional MB RAM\*
- OS X Software

Only  
**\$569**

#### Package 2:

- Internal 45 Gb IBM DeskStar 7200 rpm drive w/5 yr warranty
- 128 Mb Additional MB RAM\*
- OS X Software

Only  
**\$369**

\*For most late-model CPUs

### Cube & display w/extra RAM

Ref. G4/450 Cube upgraded to 128 mb RAM, 20 Gb drive, DVD-ROM drive and a 17" Apple Studio display.\* Extend your warranty to three years for only \$249.

**\$1445**

### Beige G3/266 Tower w/display

Get this rock-solid pre-owned G3/266 Tower complete with 128 mb RAM, 6 Gb hard drive, internal Zip drive and beautiful Apple 1710 Trinitron for one great price!

**\$945**

### iMac™ w/extra RAM & printer

Ref. Indigo 400 iMac™ DV w/ 128 mb RAM, 10 Gb drive, CD-ROM and a Canon BJC-2100 printer. Extend your warranty to 3 years for only \$149. Price after Canon mfg. rebate.

**\$885**

### G4 & 17" display w/extra RAM

New G4/400 Tower upgraded to 128 mb RAM, 20 Gb drive, DVD-ROM and a 17" ref. Apple Studio display. Extend your warranty to three years for only \$249.

**\$1685**

**LACIE**  
FireWire 45 Gb drive ref. ....\$246  
FireWire 75 Gb drive ref. ....\$446  
FireWire CD-RW 16x10x40x ....\$346  
Electron22blue II Display ref. ....\$946  
Over 100 other Lacie items on sale!

**TradeUp**  
G4  
G3  
G6  
G5  
G6  
G7  
G8  
G9  
G10  
G11  
G12  
G13  
G14  
G15  
G16  
G17  
G18  
G19  
G20  
G21  
G22  
G23  
G24  
G25  
G26  
G27  
G28  
G29  
G30  
G31  
G32  
G33  
G34  
G35  
G36  
G37  
G38  
G39  
G40  
G41  
G42  
G43  
G44  
G45  
G46  
G47  
G48  
G49  
G50  
G51  
G52  
G53  
G54  
G55  
G56  
G57  
G58  
G59  
G60  
G61  
G62  
G63  
G64  
G65  
G66  
G67  
G68  
G69  
G70  
G71  
G72  
G73  
G74  
G75  
G76  
G77  
G78  
G79  
G80  
G81  
G82  
G83  
G84  
G85  
G86  
G87  
G88  
G89  
G90  
G91  
G92  
G93  
G94  
G95  
G96  
G97  
G98  
G99  
G100  
G101  
G102  
G103  
G104  
G105  
G106  
G107  
G108  
G109  
G110  
G111  
G112  
G113  
G114  
G115  
G116  
G117  
G118  
G119  
G120  
G121  
G122  
G123  
G124  
G125  
G126  
G127  
G128  
G129  
G130  
G131  
G132  
G133  
G134  
G135  
G136  
G137  
G138  
G139  
G140  
G141  
G142  
G143  
G144  
G145  
G146  
G147  
G148  
G149  
G150  
G151  
G152  
G153  
G154  
G155  
G156  
G157  
G158  
G159  
G160  
G161  
G162  
G163  
G164  
G165  
G166  
G167  
G168  
G169  
G170  
G171  
G172  
G173  
G174  
G175  
G176  
G177  
G178  
G179  
G180  
G181  
G182  
G183  
G184  
G185  
G186  
G187  
G188  
G189  
G190  
G191  
G192  
G193  
G194  
G195  
G196  
G197  
G198  
G199  
G200  
G201  
G202  
G203  
G204  
G205  
G206  
G207  
G208  
G209  
G210  
G211  
G212  
G213  
G214  
G215  
G216  
G217  
G218  
G219  
G220  
G221  
G222  
G223  
G224  
G225  
G226  
G227  
G228  
G229  
G230  
G231  
G232  
G233  
G234  
G235  
G236  
G237  
G238  
G239  
G240  
G241  
G242  
G243  
G244  
G245  
G246  
G247  
G248  
G249  
G250  
G251  
G252  
G253  
G254  
G255  
G256  
G257  
G258  
G259  
G260  
G261  
G262  
G263  
G264  
G265  
G266  
G267  
G268  
G269  
G270  
G271  
G272  
G273  
G274  
G275  
G276  
G277  
G278  
G279  
G280  
G281  
G282  
G283  
G284  
G285  
G286  
G287  
G288  
G289  
G290  
G291  
G292  
G293  
G294  
G295  
G296  
G297  
G298  
G299  
G300  
G301  
G302  
G303  
G304  
G305  
G306  
G307  
G308  
G309  
G310  
G311  
G312  
G313  
G314  
G315  
G316  
G317  
G318  
G319  
G320  
G321  
G322  
G323  
G324  
G325  
G326  
G327  
G328  
G329  
G330  
G331  
G332  
G333  
G334  
G335  
G336  
G337  
G338  
G339  
G340  
G341  
G342  
G343  
G344  
G345  
G346  
G347  
G348  
G349  
G350  
G351  
G352  
G353  
G354  
G355  
G356  
G357  
G358  
G359  
G360  
G361  
G362  
G363  
G364  
G365  
G366  
G367  
G368  
G369  
G370  
G371  
G372  
G373  
G374  
G375  
G376  
G377  
G378  
G379  
G380  
G381  
G382  
G383  
G384  
G385  
G386  
G387  
G388  
G389  
G390  
G391  
G392  
G393  
G394  
G395  
G396  
G397  
G398  
G399  
G400  
G401  
G402  
G403  
G404  
G405  
G406  
G407  
G408  
G409  
G410  
G411  
G412  
G413  
G414  
G415  
G416  
G417  
G418  
G419  
G420  
G421  
G422  
G423  
G424  
G425  
G426  
G427  
G428  
G429  
G430  
G431  
G432  
G433  
G434  
G435  
G436  
G437  
G438  
G439  
G440  
G441  
G442  
G443  
G444  
G445  
G446  
G447  
G448  
G449  
G450  
G451  
G452  
G453  
G454  
G455  
G456  
G457  
G458  
G459  
G460  
G461  
G462  
G463  
G464  
G465  
G466  
G467  
G468  
G469  
G470  
G471  
G472  
G473  
G474  
G475  
G476  
G477  
G478  
G479  
G480  
G481  
G482  
G483  
G484  
G485  
G486  
G487  
G488  
G489  
G490  
G491  
G492  
G493  
G494  
G495  
G496  
G497  
G498  
G499  
G500  
G501  
G502  
G503  
G504  
G505  
G506  
G507  
G508  
G509  
G510  
G511  
G512  
G513  
G514  
G515  
G516  
G517  
G518  
G519  
G520  
G521  
G522  
G523  
G524  
G525  
G526  
G527  
G528  
G529  
G530  
G531  
G532  
G533  
G534  
G535  
G536  
G537  
G538  
G539  
G540  
G541  
G542  
G543  
G544  
G545  
G546  
G547  
G548  
G549  
G550  
G551  
G552  
G553  
G554  
G555  
G556  
G557  
G558  
G559  
G560  
G561  
G562  
G563  
G564  
G565  
G566  
G567  
G568  
G569  
G570  
G571  
G572  
G573  
G574  
G575  
G576  
G577  
G578  
G579  
G580  
G581  
G582  
G583  
G584  
G585  
G586  
G587  
G588  
G589  
G590  
G591  
G592  
G593  
G594  
G595  
G596  
G597  
G598  
G599  
G600  
G601  
G602  
G603  
G604  
G605  
G606  
G607  
G608  
G609  
G610  
G611  
G612  
G613  
G614  
G615  
G616  
G617  
G618  
G619  
G620  
G621  
G622  
G623  
G624  
G625  
G626  
G627  
G628  
G629  
G630  
G631  
G632  
G633  
G634  
G635  
G636  
G637  
G638  
G639  
G640  
G641  
G642  
G643  
G644  
G645  
G646  
G647  
G648  
G649  
G650  
G651  
G652  
G653  
G654  
G655  
G656  
G657  
G658  
G659  
G660  
G661  
G662  
G663  
G664  
G665  
G666  
G667  
G668  
G669  
G670  
G671  
G672  
G673  
G674  
G675  
G676  
G677  
G678  
G679  
G680  
G681  
G682  
G683  
G684  
G685  
G686  
G687  
G688  
G689  
G690  
G691  
G692  
G693  
G694  
G695  
G696  
G697  
G698  
G699  
G700  
G701  
G702  
G703  
G704  
G705  
G706  
G707  
G708  
G709  
G710  
G711  
G712  
G713  
G714  
G715  
G716  
G717  
G718  
G719  
G720  
G721  
G722  
G723  
G724  
G725  
G726  
G727  
G728  
G729  
G730  
G731  
G732  
G733  
G734  
G735  
G736  
G737  
G738  
G739  
G740  
G741  
G742  
G743  
G744  
G745  
G746  
G747  
G748  
G749  
G750  
G751  
G752  
G753  
G754  
G755  
G756  
G757  
G758  
G759  
G760  
G761  
G762  
G763  
G764  
G765  
G766  
G767  
G768  
G769  
G770  
G771  
G772  
G773  
G774  
G775  
G776  
G777  
G778  
G779  
G780  
G781  
G782  
G783  
G784  
G785  
G786  
G787  
G788  
G789  
G790  
G791  
G792  
G793  
G794  
G795  
G796  
G797  
G798  
G799  
G800  
G801  
G802  
G803  
G804  
G805  
G806  
G807  
G808  
G809  
G810  
G811  
G812  
G813  
G814  
G815  
G816  
G817  
G818  
G819  
G820  
G821  
G822  
G823  
G824  
G825  
G826  
G827  
G828  
G829  
G830  
G831  
G832  
G833  
G834  
G835  
G836  
G837  
G838  
G839  
G840  
G841  
G842  
G843  
G844  
G845  
G846  
G847  
G848  
G849  
G850  
G851  
G852  
G853  
G854  
G855  
G856  
G857  
G858  
G859  
G860  
G861  
G862  
G863  
G864  
G865  
G866  
G867  
G868  
G869  
G870  
G871  
G872  
G873  
G874  
G875  
G876  
G877  
G878  
G879  
G880  
G881  
G882  
G883  
G884  
G885  
G886  
G887  
G888  
G889  
G890  
G891  
G892  
G893  
G894  
G895  
G896  
G897  
G898  
G899  
G900  
G901  
G902  
G903  
G904  
G905  
G906  
G907  
G908  
G909  
G910  
G911  
G912  
G913  
G914  
G915  
G916  
G917  
G918  
G919  
G920  
G921  
G922  
G923  
G924  
G925  
G926  
G927  
G928  
G929  
G930  
G931  
G932  
G933  
G934  
G935  
G936  
G937  
G938  
G939  
G940  
G941  
G942  
G943  
G944  
G945  
G946  
G947  
G948  
G949  
G950  
G951  
G952  
G953  
G954  
G955  
G956  
G957  
G958  
G959  
G960  
G961  
G962  
G963  
G964  
G965  
G966  
G967  
G968  
G969  
G970  
G971  
G972  
G973  
G974  
G975  
G976  
G977  
G978  
G979  
G980  
G981  
G982  
G983  
G984  
G985  
G986  
G987  
G988  
G989  
G990  
G991  
G992  
G993  
G994  
G995  
G996  
G997  
G998  
G999  
G1000  
G1001  
G1002  
G1003  
G1004  
G1005  
G1006  
G1007  
G1008  
G1009  
G1010  
G1011  
G1012  
G1013  
G1014  
G1015  
G1016  
G1017  
G1018  
G1019  
G1020  
G1021  
G1022  
G1023  
G1024  
G1025  
G1026  
G1027  
G1028  
G1029  
G1030  
G1031  
G1032  
G1033  
G1034  
G1035  
G1036  
G1037  
G1038  
G1039  
G1040  
G1041  
G1042  
G1043  
G1044  
G1045  
G1046  
G1047  
G1048  
G1049  
G1050  
G1051  
G1052  
G1053  
G1054  
G1055  
G1056  
G1057  
G1058  
G1059  
G1060  
G1061  
G1062  
G1063  
G1064  
G1065  
G1066  
G1067  
G1068  
G1069  
G1070  
G1071  
G1072  
G1073  
G1074  
G1075  
G1076  
G1077  
G1078  
G1079  
G1080  
G1081  
G1082  
G1083  
G1084  
G1085  
G1086  
G1087  
G1088  
G1089  
G1090  
G1091  
G1092  
G1093  
G1094  
G1095  
G1096  
G1097  
G1098  
G1099  
G1100  
G1101  
G1102  
G1103  
G1104  
G1105  
G1106  
G1107  
G1108  
G1109  
G1110  
G1111  
G1112  
G1113  
G1114  
G1115  
G1116  
G1117  
G1118  
G1119  
G1120  
G1121  
G1122  
G1123  
G1124  
G1125  
G1126  
G1127  
G1128  
G1129  
G1130  
G1131  
G1132  
G1133  
G1134  
G1135  
G1136  
G1137  
G1138  
G1139  
G1140  
G1141  
G1142  
G1143  
G1144  
G1145  
G1146  
G1147  
G1148  
G1149  
G1150  
G1151  
G1152  
G1153  
G1154  
G1155  
G1156  
G1157  
G1158  
G1159  
G1160  
G1161  
G1162  
G1163  
G1164  
G1165  
G1166  
G1167  
G1168  
G1169  
G1170  
G1171  
G1172  
G1173  
G1174  
G1175  
G1176  
G1177  
G1178  
G1179  
G1180  
G1181  
G1182  
G1183  
G1184  
G1185  
G1186  
G1187  
G1188  
G1189  
G1190  
G1191  
G1192  
G1193  
G1194  
G1195  
G1196  
G1197  
G1198  
G1199  
G1200  
G1201  
G1202  
G1203  
G1204  
G1205  
G1206  
G1207  
G1208  
G1209  
G1210  
G1211  
G1212  
G1213  
G1214  
G1215  
G1216  
G1217  
G1218  
G1219  
G1220  
G1221  
G1222  
G1223  
G1224  
G1225  
G1226  
G1227  
G1228  
G1229  
G1230  
G1231  
G1232  
G1233  
G1234  
G1235  
G1236  
G1237  
G1238  
G1239  
G1240  
G1241  
G1242  
G1243  
G1244  
G1245  
G1246  
G1247  
G1248  
G1249  
G1250  
G1251  
G1252  
G1253  
G1254  
G1255  
G1256  
G1257  
G1258  
G1259  
G1260  
G1261  
G1262  
G1263  
G1264  
G1265  
G1266  
G1267  
G1268  
G1269  
G1270  
G1271  
G1272  
G1273  
G1274  
G1275  
G1276  
G1277  
G1278  
G1279  
G1280  
G1281  
G1282  
G1283  
G1284  
G1285  
G1286  
G1287  
G1288  
G1289  
G1290  
G1291  
G1292  
G1293  
G1294  
G1295  
G1296  
G1297  
G1298  
G1299  
G1300  
G1301  
G1302  
G1303  
G1304  
G1305  
G1306  
G1307  
G1308  
G1309  
G1310  
G1311  
G1312  
G1313  
G1314  
G1315  
G1316  
G1317  
G1318  
G1319  
G1320  
G1321  
G1322  
G1323  
G1324  
G1325  
G1326  
G1327  
G1328  
G1329  
G1330  
G1331  
G1332  
G1333  
G1334  
G1335  
G1336  
G1337  
G1338  
G1339  
G1340  
G1341  
G1342  
G1343  
G1344  
G1345  
G1346  
G1347  
G1348  
G1349  
G1350  
G1351  
G1352  
G1353  
G1354  
G1355  
G1356  
G1357  
G1358  
G1359  
G1360  
G1361  
G1362  
G1363  
G1364  
G1365  
G1366  
G1367  
G1368  
G1369  
G1370  
G1371  
G1372  
G1373  
G1374  
G1375  
G1376  
G1377  
G1378  
G1379  
G1380  
G1381  
G1382  
G1383  
G1384  
G1385  
G1386  
G1387  
G1388  
G1389  
G1390  
G1391  
G1392  
G1393  
G1394  
G1395  
G1396  
G1397  
G1398  
G1399  
G1400  
G1401  
G1402  
G1403  
G1404  
G1405  
G1406  
G1407  
G1408  
G1409  
G1410  
G1411  
G1412  
G1413  
G1414  
G1415  
G1416  
G1417  
G1418  
G1419  
G1420  
G1421  
G1422  
G1423  
G1424  
G1425  
G1426  
G1427  
G1428  
G1429  
G1430  
G1431  
G1432  
G1433  
G1434  
G1435  
G1436  
G1437  
G1438  
G1439  
G1440  
G1441  
G1442  
G1443  
G1444  
G1445  
G1446  
G1447  
G1448  
G1449  
G1450  
G1451  
G1452  
G1453  
G1454  
G1455  
G1456  
G1457  
G1458  
G1459  
G1460  
G1461  
G1462  
G1463  
G1464  
G1465  
G1466  
G1467  
G1468  
G1469  
G1470  
G1471  
G1472  
G1473  
G1474  
G1475  
G1476  
G1477  
G1478  
G1479  
G1480  
G1481  
G1482  
G1483  
G1484  
G1485  
G1486  
G1487  
G1488  
G1489  
G1490  
G1491  
G1492  
G1493  
G1494  
G1495  
G1496  
G1497  
G1498  
G1499  
G1500  
G1501  
G1502  
G1503  
G1504  
G1505  
G1506  
G1507  
G1508  
G1509  
G1510  
G1511  
G1512  
G1513  
G1514  
G1515  
G1516  
G1517  
G1518  
G1519  
G1520  
G1521  
G1522  
G1523  
G1524  
G1525  
G1526  
G1527  
G1528  
G1529  
G1530  
G1531  
G1532  
G1533  
G1534  
G1535  
G1536  
G1537  
G1538  
G1539  
G1540  
G1541  
G1542  
G1543  
G1544  
G1545  
G1546  
G1547  
G1548  
G1549  
G1550  
G1551  
G1552  
G1553  
G1554

# Scripter 2

NEW!  
Version 2.2

with **ScriptBASE**

**Tap the power of AppleScript  
with Main Event's Scripter!**

**For professionals and novices  
Webmasters and solution providers**

**No matter what your experience,  
Scripter makes it easier!**

## BEGINNER AT SCRIPTING?

- Unique command-builders help you learn to assemble commands
- Built-in tools for experimenting
- Shortcuts to speed scripting

## EXPERIENCED WITH APPLESCRIPT?

- Unmatched single-step interactive debugging
- Watch and modify global and local variables while stepping
- Debug messages sent to script applications
- Includes ScriptBase to integrate scripting scenarios

*Scripter received MacWeek's 5-diamond rating – Twice!  
"Scripter's virtuoso display of AppleScripting savvy and  
debugging wizardry make it the best environment we've  
found for learning or creating AppleScript scripts."*

—MacWeek

**Make AppleScript and your  
applications work for you!**



Main Event  
PO Box 21470  
Washington, DC 20009  
Tel: 202-298-9595  
Sales: 800-616-8320  
info@mainevent.com  
www.mainevent.com

### Listing 10: Adding samples to the icon sprite movie

```
QTSprites_AddIconMovieSamplesToMedia (Media theMedia)
{
    QTAtomContainer    mySample = NULL;
    QTAtomContainer    mySpriteData = NULL;
    RGBColor           myKeyColor;
    Point              myLocation;
    short              isVisible;
    short              myLayer, myIndex, myCount;
    OSErr              myErr = noErr;

    // create a new, empty key frame sample
    myErr = QTNewAtomContainer(&mySample);
    if (myErr != noErr)
        goto bail;

    myKeyColor.red = myKeyColor.green = myKeyColor.blue =
        0xffff; // white

    // add images to the key frame sample
    SpriteUtils_AddPictImageToKeyFrameSample(mySample,
        kOldQTIconID, &myKeyColor, 1, NULL, NULL);
    SpriteUtils_AddPictImageToKeyFrameSample(mySample,
        kNewQTIconID, &myKeyColor, 2, NULL, NULL);

    // add the initial sprite properties to the key frame sample
    myErr = QTNewAtomContainer(&mySpriteData);
    if (myErr != noErr)
        goto bail;

    // the QT icon sprite
    myLocation.h = 32;
    myLocation.v = 32;
    isVisible = true;
    myLayer = -1;
    myIndex = 1;

    SpriteUtils_SetSpriteData(mySpriteData, &myLocation,
        &isVisible, &myLayer, &myIndex, NULL, NULL, NULL);
    SpriteUtils_AddSpriteToSample(mySample, mySpriteData,
        kQTIconSpriteAtomID);
    SpriteUtils_AddSpriteSampleToMedia(theMedia, mySample,
        kSpriteMediaFrameDurationIcon, true, NULL);

    // add a few override samples to change the icon's location and image
    for (myCount = 1; myCount <= kNumOverrideSamples;
        myCount++) {
        QTRemoveChildren(mySample, kParentAtomIsContainer);
        QTRemoveChildren(mySpriteData, kParentAtomIsContainer);

        // every frame, bump the icon's location
        myLocation.h += 2;

        // change icon half way thru
        if (myCount == kNumOverrideSamples / 2)
            myIndex = 2;

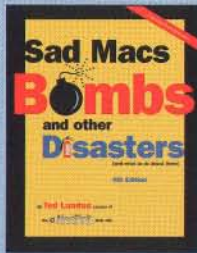
        SpriteUtils_SetSpriteData(mySpriteData, &myLocation,
            NULL, NULL, &myIndex, NULL, NULL, NULL);
        SpriteUtils_AddSpriteToSample(mySample, mySpriteData,
            kQTIconSpriteAtomID);
        SpriteUtils_AddSpriteSampleToMedia(theMedia, mySample,
            kSpriteMediaFrameDurationIcon, false, NULL);
    }

bail:
    if (mySample != NULL)
        QTDisposeAtomContainer(mySample);

    if (mySpriteData != NULL)
        QTDisposeAtomContainer(mySpriteData);
}
```

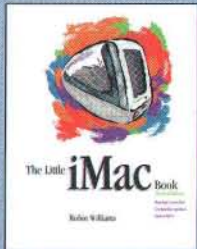
# Peachpit Press

## Get the most from your Mac



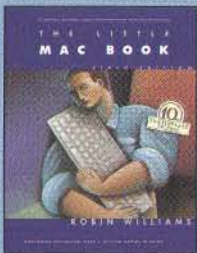
### Sad Macs, Bombs, and Other Disasters, 4th Edition

By Ted Landau  
ISBN 0-201-69963-X  
980 pages • \$34.99 u.s.



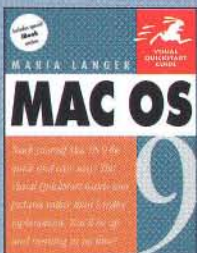
### The Little iMac Book, Second Edition

By Robin Williams  
ISBN 0-201-70446-3  
312 pages • \$17.99 u.s.



### The Little Mac Book, Sixth Edition

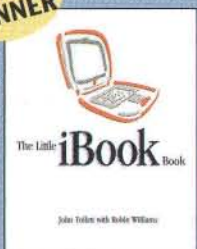
By Robin Williams  
ISBN 0-201-35433-0  
448 pages • \$19.99 u.s.



### Mac OS 9: Visual QuickStart Guide

By Maria Langer  
ISBN 0-201-70004-2  
376 pages • \$17.99 u.s.

AWARD  
WINNER



### The Little iBook Book

By John Tollett and Robin Williams  
ISBN 0-201-70093-X  
232 pages • \$18.99 u.s.

# Great Books!

## The hottest graphics and Web books at great prices!

NEW



### Flash 5 for Windows and Macintosh: Visual QuickStart Guide

By Katherine Ulrich  
ISBN 0-201-71614-3  
544 pages • \$21.99 u.s.

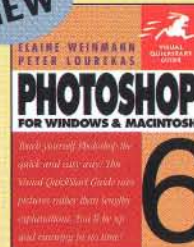
BEST  
SELLER



### HTML for the World Wide Web, Fourth Edition: Visual QuickStart Guide

By Elizabeth Castro  
ISBN 0-201-35493-4  
384 pages • \$19.99 u.s.

NEW



### Photoshop 6 for Windows and Macintosh: Visual QuickStart Guide

By Elaine Weinmann and Peter Lourekas  
ISBN 0-201-71309-8  
448 pages • \$21.99 u.s.



Peachpit Press

Peachpit Press books are available at quality bookstores everywhere. For a list of bookstores in your area see [www.peachpit.com/order/stores.html](http://www.peachpit.com/order/stores.html)

Listing 11 shows our complete definition of QTSprites\_AddPenguinMovieSamplesToMedia.

#### Listing 11: Adding samples to the penguin sprite movie

```
QTSprites_AddPenguinMovieSamplesToMedia
void QTSprites_AddPenguinMovieSamplesToMedia (Media theMedia)
{
    QAtomContainer    mySample = NULL;
    QAtomContainer    mySpriteData = NULL;
    RGBColor          myKeyColor;
    Point             myLocation;
    short             isVisible;
    short             myLayer, myIndex, myCount;
    ModifierTrackGraphicsModeRecord
                    myGraphicsMode;
    OSErr             myErr = noErr;

    // create a new, empty key frame sample
    myErr = QTNewAtomContainer(&mySample);
    if (myErr != noErr)
        goto bail;

    myKeyColor.red = myKeyColor.green = myKeyColor.blue =
        0xffff;    // white

    // add images to the key frame sample
    SpriteUtils_AddPICTImageToKeyFrameSample(mySample,
        kPenguinPictID, &myKeyColor,
        kPenguinImageIndex, NULL, NULL);

    // add the initial sprite properties to the key frame sample
    myErr = QTNewAtomContainer(&mySpriteData);
    if (myErr != noErr)
        goto bail;
```

```
// the penguin sprite
myLocation.h = 0;
myLocation.v = 0;
isVisible = true;
myLayer = -1;
myIndex = 1;

// set the initial blend amount (0 = fully transparent; 0xffff = fully opaque)
myGraphicsMode.graphicsMode = blend;
myGraphicsMode.opColor.red = 0;
myGraphicsMode.opColor.green = 0;
myGraphicsMode.opColor.blue = 0;

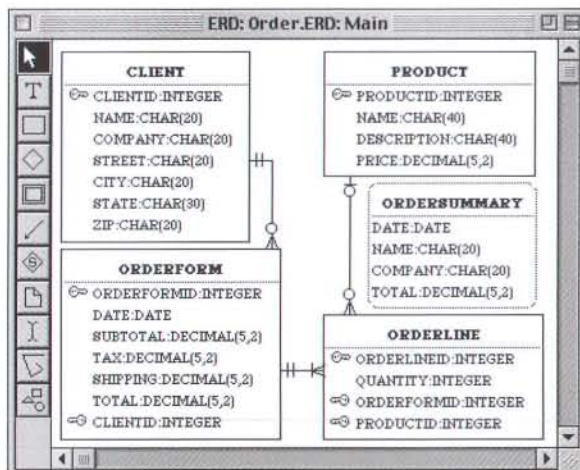
SpriteUtils_SetSpriteData(mySpriteData, &myLocation,
    &isVisible, &myLayer, &myIndex, &myGraphicsMode,
    NULL, NULL);
SpriteUtils_AddSpriteToSample(mySample, mySpriteData,
    kPenguinSpriteAtomID);
SpriteUtils_AddSpriteSampleToMedia(theMedia, mySample,
    kSpriteMediaFrameDurationPenguin, true, NULL);

// add a few override samples to change the penguin's opacity
for (myCount = 1; myCount <= kNumOverrideSamples;
    myCount++) {
    QTRemoveChildren(mySample, kParentAtomIsContainer);
    QTRemoveChildren(mySpriteData, kParentAtomIsContainer);

    // every frame, bump the penguin's opacity
    myGraphicsMode.graphicsMode = blend;
    myGraphicsMode.opColor.red = (myCount - 1) *
        (0xffff / kNumOverrideSamples - 1);
    myGraphicsMode.opColor.green = (myCount - 1) *
        (0xffff / kNumOverrideSamples - 1);
    myGraphicsMode.opColor.blue = (myCount - 1) *
        (0xffff / kNumOverrideSamples - 1);
```

## The best design tool just got better!

- ✓ Complete UML Design for OO Systems
- ✓ Structure Analysis & Design
- ✓ Real-Time, Multitask Design
- ✓ Requirements Traceability
- ✓ Code Generation and Reengineering
- ✓ Scriptable HTML Reports
- ✓ Rich Data Models & SQL Generation



## SQL database design in MacA&D

Software for the business world revolves around SQL database systems like Oracle, SQL Server, Sybase, DB2, Informix and InterBase. Database designers can create logical and physical data models with MacA&D, then generate SQL for tables, views, constraints, assertions, triggers, indexes, procedures and other SQL elements.

Designers can toggle between logical and physical models and show tables with full attribute lists, just primary and foreign keys or customized to show specific attributes.

MacTranslator generates a data model from existing SQL. MacA&D and MacTranslator use namespaces to simplify team development, enable powerful organizational and reporting features, partition complex systems and map design elements to specific SQL schemas.

**Excel Software**  
Software Engineering Made Easy

505-771-3719

Fax: 505-771-3718

**www.excelsoftware.com**

Information on software models, design methods and software engineering tools for Macintosh and Windows 95/NT computers.

MacA&D, MacTranslator, WinA&D, WinTranslator and QuickCRC are trademarks of Excel Software. © 2001. Other brand and product names are trademarks of their respective holders.



# Programming Doesn't Have To Be This Difficult. It's REALbasic!

**NEW**  
Version!  
REALbasic 2.1\*  
**FREE**  
Update!

**REALbasic is the award-winning, visual, object-oriented BASIC development environment for the Macintosh.**

Use REALbasic's visual interface builder and platform-independent language to build native, compiled—not interpreted—professional quality applications in a fraction of the time it would take in C/C++. Because our language is platform-independent you only need to write a single set of code to create applications for both Macintosh and Windows. Leverage your C/C++ experience to extend REALbasic's capabilities using shared libraries, Mac OS Toolbox and Win32 API calls or by writing cross-platform REALbasic plug-ins. Create prototypes, Internet applications, database front-ends, even games with REALbasic! Its OOP language employs everything

you'd expect from a modern development environment—methods, properties, classes, subclassing, inheritance, constructors and destructors, virtual methods, and more. The IDE supports multi-threading, extensibility, TCP/IP controls, plus multimedia and QuickTime tools, and support for standards such as SQL, ODBC, AppleScript, and XCMDs. You can even import Visual Basic forms and modules.

**Go to [www.realbasic.com](http://www.realbasic.com) NOW to download a FREE trial version or call 512.263.1233.**



## REALbasic®



2000 Runner-Up - Best Macintosh User Experience

\*Free update for all owners of REALbasic 2.0 and above. REALbasic and the REALbasic logo are trademarks of REAL Software, Inc. Apple and the Apple logo are trademarks of Apple Computer, Inc., registered in the U.S., used with permission. All other trademarks are the property of their respective owners.

```

SpriteUtils_SetSpriteData(mySpriteData, NULL, NULL, NULL,
    NULL, &myGraphicsMode, NULL, NULL);
SpriteUtils_AddSpriteToSample(mySample, mySpriteData,
    kPenguinSpriteAtomID);
SpriteUtils_AddSpriteSampleToMedia(theMedia, mySample,
    kSpriteMediaFrameDurationPenguin, false, NULL);
}

bail:
    if (mySample != NULL)
        QTDisposeAtomContainer(mySample);

    if (mySpriteData != NULL)
        QTDisposeAtomContainer(mySpriteData);
}

```

For the definition of `QTSprites_AddSpaceMovieSamplesToMedia`, see the file `QTSprites.h`. The basic strategy is the same, but the increased complexity of the space movie gives a considerably longer function.

### HIT TESTING

As I mentioned earlier, QuickTime sprites can be interactive. That is to say, we can create movies with sprite tracks whose sprites respond to user actions like mouse movements and mouse button clicks. When we consider wired sprites in a future article, we'll investigate the full power of this interactivity. In the meantime, let's take a look at a more limited form of interactivity supported by non-wired sprites, the ability to determine whether the user has clicked on a sprite (also called *hit testing*).

Programmatically finding clicks on a sprite is a two-stage process. First, we need to determine when the user has clicked the mouse button inside of the movie rectangle. Then we need to determine whether that mouse click was on top of a sprite. For the first task, we can add a case to the `switch` statement in our movie controller action filter function `QTAApp_MCAActionFilterProc`, looking for the `mcActionMouseDown` movie controller action (as shown in Listing 12).

#### Listing 12: Detecting clicks in the movie window

```

switch (theAction) {
    // handle window resizing
    case mcActionControllerSizeChanged:
        QTFrame_SizeWindowToMovie(myWindowObject);
        break;

    // handle idle events
    case mcActionIdle:
        QTApp_Idle((**myWindowObject).fWindow);
        break;

    // handle mouse-down events
    case mcActionMouseDown:
        isHandled = QTSprites_HitTestSprites(myWindowObject,
            (EventRecord *)theParams);

        break;

    default:
        break;
}

```

The movie controller issues an `mcActionMouseDown` action whenever it receives a mouse-down event that's inside the movie rectangle. The parameter data passed to our action filter function (in `theParams`) is a pointer to the event record for that mouse-down event. As you can see, our filter function simply calls the application-defined function `QTSprites_HitTestSprites` to see whether the click is on a sprite and, if so, to react appropriately. (For more information about movie controller action filter functions, see "QuickTime 101" in *MacTech*, January 2000.)

The sprite media handler supplies the `SpriteMediaHitTestOneSprite` and `SpriteMediaHitTestAllSprites` functions, which we can use to determine whether the user has clicked on a sprite. For present purposes, we'll use `SpriteMediaHitTestAllSprites`, which looks at each one of the sprites in a sprite track to see whether it is currently at a specified location. `SpriteMediaHitTestAllSprites` is declared essentially like this:

```

ComponentResult SpriteMediaHitTestAllSprites
    (MediaHandler mh, long flags, Point loc,
     QAtomID *spriteHitID);

```

If a sprite is situated at the specified location in the sprite track associated with the specified sprite media handler, then `SpriteMediaHitTestAllSprites` returns the ID of that sprite in the `spriteHitID` parameter. If more than one sprite is situated at that location, then `spriteHitID` is set to the ID of the frontmost sprite (that is, the sprite with the lowest layer number). If no sprite is situated at that location, then `spriteHitID` is set to 0.

By default, the `loc` parameter should be the location of the mouse click, in coordinates local to the sprite track. However, the event record passed to our movie controller action filter function contains the location of the mouse click in global coordinates. Rather than bother with converting the global location to a local position, we can add `spriteHitTestLocInDisplayCoordinates` to the `flags` parameter, to indicate that the `loc` parameter is in global coordinates. These flags are understood by `SpriteMediaHitTestAllSprites`:

```

enum {
    spriteHitTestBounds          = 1L << 0,
    spriteHitTestImage           = 1L << 1,
    spriteHitTestInvisibleSprites = 1L << 2,
    spriteHitTestIsClick          = 1L << 3,
    spriteHitTestLocInDisplayCoordinates = 1L << 4
};

```

If you want to accept clicks anywhere within the rectangular bounding box of a sprite, add in the `spriteHitTestBounds` flag. If, conversely, you want to accept clicks only on a non-transparent part of a sprite, add in the `spriteHitTestImage` flag. You must specify one or the other of these two flags, or else no hit-testing will occur. Setting both of these flags is tantamount to setting only `spriteHitTestImage`. (Earlier versions of QuickTime required you to set both flags if you wanted image testing, but current versions do not.)



# DAVE?<sup>TM</sup> It's totally cool!

How cool would it be for your Macs and PCs to become best friends? To share text and graphics files and postscript printers across a network with no barriers. Get DAVE, the "totally cool" file share solution from Thursby Software. DAVE installs on your Mac. It's fast, secure and simple to use... perfect for sharing with Windows 95/98/Me/NT/2000 and soon, for Mac OS X.

Download a **FREE EVALUATION** today.



*The File Share Folks<sup>TM</sup>*

[www.thursby.com](http://www.thursby.com)

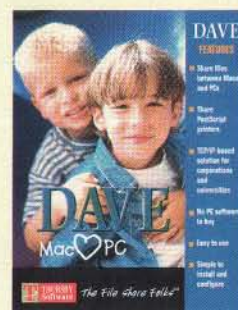
© 2001 Thursby Software Systems, Inc.



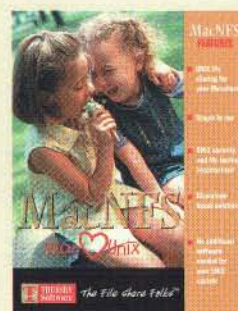
**THURSBY SOFTWARE** has the  
perfect file share solution to  
meet your needs.



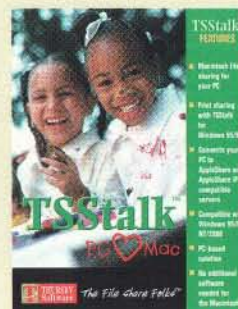
**The ideal solution for the small  
office where Macs & PCs need  
to share files.**



**The fast, easy way to share  
files and printers between  
PCs and Macintosh systems.**



**Achieve true NFS connectivity  
between your Mac and  
UNIX systems.**



**Windows users can access  
files on any Mac workstation  
or AppleShare server.**

By default, `SpriteMediaHitTestAllSprites` and `SpriteMediaHitTestOneSprite` test only visible sprites for hits. If you want to test invisible sprites as well, set the `spriteHitTestInvisibleSprites` flag. If you want to pass the mouse click onto the codec that is rendering the sprite image, then specify the `spriteHitTestIsClick` flag; this is currently useful only with the Ripple codec.

In `QTSprites_HitTestSprites`, we want to test both visible and invisible sprites for hits, and we want to test only within the non-transparent portions of a sprite image. So we'll specify our flags parameter like this:

```
myFlags = spriteHitTestImage |
           spriteHitTestLocInDisplayCoordinates |
           spriteHitTestInvisibleSprites;
```

Now, what will we do when we detect a click on a sprite? In theory, we can do anything we want. We've got the entire QuickTime API at our disposal, so we could do some neat things like launch the user's web browser and navigate to a specific page, or download a file from a remote location, or even build a QuickTime movie. Even just within the context of our sprite track we could do some rather interesting stuff, like moving the sprite to a new location in the sprite track, changing its image index, and so forth. For simplicity, we'll limit ourselves to changing the sprite's visibility state: if the sprite is currently visible, we'll make it invisible (and vice versa).

```
SpriteMediaGetSpriteProperty(myHandler, myAtomID,
```

```
kSpritePropertyVisible, (void *)&isVisible);
SpriteMediaSetSpriteProperty(myHandler, myAtomID,
                             kSpritePropertyVisible, (void *)&isVisible);
```

Listing 13 shows our complete definition of `QTSprites_HitTestSprites`.

### Listing 13: Hit testing sprites

```
QTSprites_HitTestSprites
Boolean QTSprites_HitTestSprites
    (WindowObject theWindowObject, EventRecord *theEvent)
{
    ApplicationDataHdl myAppData = NULL;
    MediaHandler        myHandler = NULL;
    Boolean             isHandled = false;
    long               myFlags = 0L;
    QTAtomID           myAtomID = 0;
    Point              myPoint;
    ComponentResult     myErr = noErr;

    myAppData = (ApplicationDataHdl)
        QTFrame_GetAppDataFromWindowObject(theWindowObject);
    if (myAppData == NULL)
        goto bail;

    if (theEvent == NULL)
        goto bail;

    // make sure that the click is in our window
    if ((**theWindowObject).fWindow !=
        QTFrame_GetFrontMovieWindow())
        goto bail;

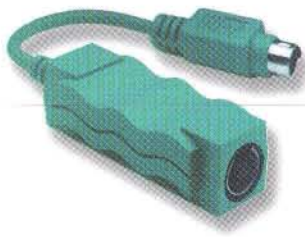
    myHandler = (**myAppData).fSpriteHandler;
    myFlags = spriteHitTestImage |
              spriteHitTestLocInDisplayCoordinates |
```

# Software Protection

## WIBU-KEY

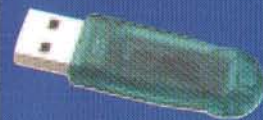
## USB and ADB for Macintosh

WIBU-KEY is the most advanced Mac copy protection & license management solution available today. The 3.0 Mac driver is available for both MacOS 9 and MacOS X.



The USB hardware is the same for the Macintosh and the PC. Inventory one type of key and deliver it with Macintosh and Windows versions of your software. You can even manage network licenses across heterogeneous networks.

WIBU-SYSTEMS has been a member of USB Implementers Forum since 1997. As the first copy protection product certified by the USB-IF, the WIBU-BOX/U successfully completed all USB Compliance tests and is officially included in the USB Integrators List.



**Best Service**  
**New Technologies**  
**High Quality Products**  
**Order Your Test Kit**  
**1-800-986-6578**  
**www.wibu.com**

**GRIFFIN**  
 TECHNOLOGIES

USA, Canada: Griffin Technologies, LLC  
 916 Massachusetts Street, Lawrence, KS 66044  
 phone: (785) 832-2070 · fax: (785) 832-8787  
 email: sales@griftech.com · www.griftech.com

**CeBIT**  
 HANNOVER  
 22. - 28. 3. 2001  
 Hall 13 Booth E66

Argentina, Australia, Belgium,  
 Canada, Croatia, Denmark, Finland,  
 France, Germany, Great Britain, Hungary,  
 Japan, Jordan, Korea, Lebanon, Mexico,  
 Netherlands, Spain, Portugal, Syria, USA

International: WIBU-SYSTEMS AG  
 Rueppurrer Strasse 52-54 · D-76137 Karlsruhe  
 phone: +49-721-93172-0 · fax: +49-721-93172-22  
 email: info@wibu.com · http://www.wibu.com

**WIBU**  
**SYSTEMS**  
 Quality the World Trusts



# Control your web database development destiny. today.

Today is the day you take control of your destiny  
as a professional application developer.  
**No excuses.** You're going to do it today, **Period.**

Your new destiny begins with a look at the development tools you're using. Are they open? flexible? scalable? Do you always worry that your tools can't accomplish the job? If your tools limit your ability to deliver solutions, how can you realize your true potential?

The tool to help you get the most out of your potential is 4th Dimension. The 4th Dimension (4D) product line is a carefully integrated, yet open development environment that offers unprecedented flexibility and scalability. With the release of 4D version 6.7 there's never a need to worry about the ability to accomplish the job at hand. 4D supports a massive range of solutions including web access, SSL, XML, WAP, FTP, Email and more. When your client asks for connectivity to other RDBMS systems like ORACLE™ or

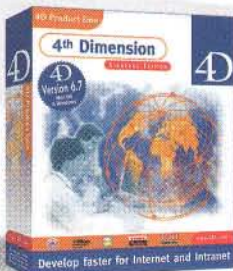
MS SQL Server™ you simply won't have to worry - 4D has been providing connectivity for over a decade. What if you could offer that same client security for the data in the database in the event the database was stolen? With 4D Version 6.7 we've pioneered the use of SSL-based technology to shore-up and encrypt the data in the database. Yet there's so much more.

Are you ready to take control, to deliver versatile solutions that meet or exceed your client expectations? You can take that first step by downloading the demo of 4D Version 6.7. The demo includes a tutorial that quickly immerses you in 4D and targets you in the right direction.

Here's to your new destiny using 4D.

[www.4D.com/destiny](http://www.4D.com/destiny)

1.800.881.3466 opt 3



## 4th Dimension v6.7

RAD • RDBMS • CGI • HTTP • SSL

```

        spriteHitTestInvisibleSprites;
myPoint = theEvent->where;

myErr = SpriteMediaHitTestAllSprites(myHandler, myFlags,
                                     myPoint, &myAtomID);
if ((myErr == noErr) && (myAtomID != 0)) {
    Boolean    isVisible;

    // the user has clicked on a sprite;
    // for now, we'll just toggle the visibility state of the sprite
    SpriteMediaGetSpriteProperty(myHandler, myAtomID,
                                kSpritePropertyVisible, (void
*)&isVisible);
    SpriteMediaSetSpriteProperty(myHandler, myAtomID,
                                kSpritePropertyVisible, (void
*)!isVisible);

    isHandled = true;
}

bail:
    return(isHandled);
}

```

You'll notice that we need to make sure that the window associated with the window object passed to `QTSprites_HitTestSprites` is the frontmost movie window, by executing these lines of code:

```

if ((*theWindowObject).fWindow !=
    QTFrame_GetFrontMovieWindow())
    goto bail;

```

This is because, when we are handling an event on Macintosh computers, we call `MCIsPlayerEvent` for each open movie controller, until we find one that handles the event. It's possible to have two or more overlapping sprite movies such that a mouse click does not hit a sprite in the frontmost movie window but does hit one in an overlapped window. We can avoid unexpected behaviors by limiting our hit testing to the frontmost movie window. (On Windows, this additional check is unnecessary but harmless.)

## CONCLUSION

In this article, we've seen how to create and work with sprite tracks in QuickTime movies. Sprite tracks are remarkably easy to create. It's really just an exercise in creating atom containers and adding them as samples to a sprite media. Keep in mind that sprite media data isn't just a collection of pixels that are copied from the movie file onto the screen. Rather, sprites are distinct objects, with properties that can be changed dynamically to animate the sprites. This is what accounts for the drastic size differences between sprite and non-sprite versions of a movie. This is also what accounts for our ability to interact with individual sprites (for instance, our ability to hit test mouse clicks on sprites).

In the next article, we'll continue our investigation of sprites. Among other things, we'll see how to use a video track as the source of a sprite's image data, and we'll see how to use modifier tracks to animate sprites. Believe it or not, using modifier tracks instead of override samples will result in even further size reductions for our sprite movie files. Stay tuned!

## ACKNOWLEDGEMENTS AND REFERENCES

The utilities in the file `SpriteUtilities.c` were originally written by Sean Allen; I have taken the liberty of reworking them to bring the general programming style into conformance with the rest of the sample code we've encountered in this series of articles. The code for building the space movie is based on an existing sample code package, `MakeSpriteMovie.c`, also by Sean Allen.

The definitive API reference for QuickTime sprites and wired actions is the book *QuickTime Wired Movies And Sprite Animation* by Tom Maremaa (downloadable from <http://developer.apple.com/techpubs/quicktime/qtdevdocs/RM/pdf/frame.htm>).

MT



# C++ Developer's Kit

## PROFESSIONAL

### Advanced Tools for ANSI C++ Development

Suite of tools and reference collection showing developers how to create better ISO/IEC C++ code using UML & C++. It includes libraries of useful programming snippets and classes that you can use in your own projects to create more efficient powerful and reliable Mac applications.



"C++ Developer's Kit offers a complete set of tools to build software conforms to ANSI/ISO C++ standards. The package also helps solve problems that might emerge in development. The software explains the fundamentals of UML and OO development with a variety of models and applications."

Institute of Electrical & Electronics Engineers

### From C to C++ more effectively!

C++ Developer's Kit Professional is a comprehensive C++ development kit for developers who need powerful C++ and UML features. This new development kit includes ISO/IEC C++ based examples for CodeWarrior, MPW, Symantec C++. This kit offers examples, projects, classes, functions libraries, templates, algorithms, tools and much more. It also provides over 250,000 code lines, the complete CAD++ library with classes and functions to design CAD applications, Garbage Collection-based classes, nested classes, UML/C++ models and more!

#### INSIDE THE KIT

3 CDs  
2 Manuals  
1 UML/C++ Quick Reference

#### INCLUDES

C++ guide  
C++ tools  
C++ snippets  
C++ projects  
C++ classes  
C++ Web sites  
UML standard  
UML seminar  
UML models  
STL library  
OOP topics  
Software Reuse

COMPUTER

**C++ developer's news**  
Technical News for Software Developers

Try out free C++ sample code at: <http://www.technosoftweb.com/C++>

**CodeWarrior**  
LITE version INSIDE

**TECHNOFT**

mailto: [orders@technosoftweb.com](mailto:orders@technosoftweb.com)

[www.technosoftweb.com](http://www.technosoftweb.com)

**DHL**  
Express service

**FREE shipping**  
(For Europe-CEE, USA and CANADA)

Just \$189 from

**DEPOT**  
800-828-8888  
www.technosoftweb.com

# LA to New York

(and Seattle and Miami)

in 150 milliseconds or less\*

▼ Edit View Special

Your Application 96N

Multi-point Meetings

Real-time Collaboration

Virtual Whiteboard™

Instant Assessment™

Best-quality video

Multimedia Presentation



Call for a TeraMedia® demo

**TeraGlobal**  
COMMUNICATIONS

(858) 404.5500 ext 5660  
www.teraglobal.com  
sales@teraglobal.com



Interact. Collaborate. It's highest-quality multi-point video conferencing and true real-time multimedia collaboration. It's all done in software. Any number of users. Over any distance. At any time. In real time. It flies, you don't.

\* Actual video latency varies with bandwidth. In most cases, TeraMedia® latency is below the threshold of human perception.

**TeraMedia**®  
by TeraGlobal



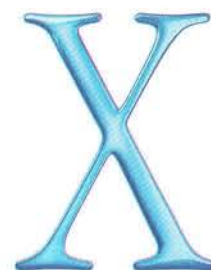
# Carbon: An Essential Element of Mac OS X

One of the things that makes Mac OS X unique is its integration of five separate application runtime environments: Carbon, Cocoa, Classic, Java, and BSD. Never before has an operating system offered developers so many options. This series of articles will give you the information you need to make an informed decision as to which of these environments is right for your application on Mac OS X. This month, I'm going to talk about Carbon and what it offers you as a Macintosh developer.

## A Brief History of Carbon

Carbon represents the evolution of the Mac OS API, from a single-user, cooperatively scheduled system, to a modern multi-user system with memory protection and preemptive multitasking. Carbon

sprang from Apple's desire to provide a gentle migration path for developers, while at the same time delivering all the performance, features, and reliability that Mac OS X has to offer. About 70 percent of the classic Mac OS APIs are fully supported, and Carbon adds many new APIs specifically to take advantage of Mac OS X.



## What Diamonds Are Made Of

Carbon offers a wealth of features to support and accelerate your Mac OS X development efforts. Whether you're new to the platform, or an old-timer, here are some of the reasons why you'll find Carbon a solid foundation for your software product:

- *Broad Language Support*

Apple provides interfaces, compilers, and tools for Carbon programming in C and C++, while other languages such as BASIC and FORTRAN are offered by other tool vendors.

- *Excellent Tools*

As we covered in last month's article, *Tools for the Trade*, you can choose from a range of excellent tools including Metrowerks CodeWarrior, Real Software's REALbasic, Absoft Pro Fortran, and Apple's own IDE combo, Project Builder and Interface Builder.

- *Choice of Frameworks*

The two most popular object-oriented application frameworks for Macintosh are now Carbonized: MacApp (part of Apple's suite of C++ frameworks) and Metrowerks PowerPlant (version 2.1 or later). Basing your application on one of these frameworks can greatly simplify your work.

- *Support for Mac OS 9*

Because the majority of Carbon APIs are also supported on Mac OS 9, Carbon provides a solution to the problem of how to move forward with Mac OS X while continuing to support those people who are going to wait a bit longer to upgrade. If backward compatibility is important for your product, you can create a single binary that will run on any version of the Mac OS, from Mac OS 8.1 through Mac OS X.

## ADC Programs and Mac OS X

As Steve Jobs, Apple's CEO, announced at Macworld San Francisco in January, Mac OS X will be released on March 24, 2001. All ADC Members who receive a monthly mailing will be automatically sent Mac OS X and the Mac OS X Developer Tools on CD-ROM upon release.

If you are getting your hardware or software products ready for Mac OS X, be sure to take advantage of our numerous ADC Member Discounts for Cocoa and Carbon porting, Aqua integration, driver development services, and more. Mac customers will be looking for hardware and software products that are built for Mac OS X – make sure yours are ready! For the complete list of all ADC Member Discounts, please see:

<http://developer.apple.com/business/>

For more information about ADC Programs, please see:

<http://developer.apple.com/membership/>

## Getting Started

If you're starting from scratch, and you want to program in C or C++, you should really consider basing your Carbon application on a framework like PowerPlant or MacApp, which provide for rapid development and easy maintenance. These frameworks can also make it easier to port your code to or from the Microsoft Win32 API, if that's a consideration for you. REALbasic and Absoft Pro Fortran also offer cross-platform solutions.

For an excellent introduction to Carbon programming, grab a copy of *Moon Travel Tutorial: Creating a Carbon Application*. This little gem takes you through the entire process of building a pure Carbon app—one that doesn't rely on any legacy technologies. You'll also learn how to use Apple's Project Builder IDE, and see how easy it is to design your menus, controls, dialogs, and windows using Interface Builder. You can download it from:

<http://developer.apple.com/techpubs/macosx/Carbon/pdf/MoonTravel.pdf>

The best place to begin exploring Carbon is the Carbon Developer home page, where you'll find links to the latest Carbon software development kits, documentation, tools and other resources:

<http://developer.apple.com/macosx/carbon/>

If you've already got a Macintosh application that you haven't yet Carbonized, it's time to get started! One of the great things about Carbon is that you can adopt it incrementally. The basic steps necessary to get an application up and running on Mac OS X typically take just a few days, but to completely Carbonize a large application you should expect to spend a month or more, especially if you need to support Mac OS 9 or prior versions.

The ease of Carbonizing an existing application is directly related to how closely you've followed Apple's recommended programming practices in the past, the degree to which you've adopted recent APIs such as Navigation Services and the Appearance Manager, and whether your code already compiles using the latest Universal Interfaces. If your code base is up-to-date, and you're not relying on undocumented data structures or unsupported features, adopting Carbon is relatively simple.

Before changing a line of code, however, you should run your application through Carbon Dater, Apple's Carbon compatibility tester. Available on the Carbon Developer web site, Carbon Dater is a helpful tool that examines your application and issues a report listing all the Mac OS functions you're using that need to be updated for Carbon compatibility. For example, if you're calling `InvalidRect`, the report will explain that the function is not supported in Carbon, and recommend that you use `InvalidWindowRect` instead.

With the Carbon Dater report as your guide, you can begin the Carbonization process by searching your sources for unsupported functions and replacing them with their Carbon equivalents. While I can't cover the entire Carbonization process in detail here, you'll

find a complete description in the *Carbon Porting Guide*, which you can download from here:

<http://developer.apple.com/techpubs/macosx/Carbon/pdf/CarbonPortingGuide.pdf>

Even if you're not porting, I recommend reading the Carbon Porting Guide for its overview of new Carbon technologies, optimization strategies, and instructions for building Carbon applications using CodeWarrior, Project Builder, and the Macintosh Programmer's Workshop (MPW). *Technical Note 2003: Moving Your Code to Mac OS X*, is another great resource. It highlights all the major considerations for developers transitioning from classic Mac OS to Carbon, and provides pointers to additional documentation on the web:

<http://developer.apple.com/technotes/tn/tn2003.html>

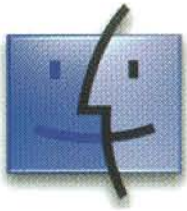
Once your application is Carbonized, you'll want to make sure it looks great on Mac OS X. This means putting the polish on your interface by giving it an Aqua tune-up. The Aqua Human Interface Guidelines contains all the specifications and recommendations for designing Aqua-compliant applications and icons:

<http://developer.apple.com/techpubs/macosx/SystemOverview/AquaGuidelines.pdf>

Continued on page 61 

## Project Builder Tips & Tricks

- To get a pop-up menu containing all the functions in your source files, select "Index Project" from the Project menu. This also enables definition searches from the batch find panel. And, the counterpart button (the third button from the right in the navigation bar) becomes a pop-up menu that shows files that are imported by or dependent on the current file.
- The inspector panel, available from the Project menu, can be used to modify more than one file at a time. For instance, if you want to set the tab width for a number of files all at once, select them in the Groups and Files pane, bring up the Inspector, change the tab width, and hit return.
- To set a breakpoint on a line of code, just click in the column to the left of that line of code. Clicking again enables and disables the breakpoint. To remove the breakpoint, drag it off the column. You can also move a breakpoint to a different line by dragging it.
- To rename a file or group, option-click its name from the Groups and Files pane.



## New Mac OS X Related Releases

The following software is available from the Download Software area of the ADC Member Site at:

<http://connect.apple.com/>

- **CarbonLib 1.2.5b2 SDK**

The latest prerelease version of the CarbonLib 1.2.5 software development kit provides all the files needed to begin Carbon development.

<http://connect.apple.com/>

- **OpenGL SDK 1.2**

This software development kit is for adding OpenGL support to your application. This implementation provides several features and performance enhancements.

<http://developer.apple.com/opengl/downloads.html>

### Developer Documentation

The following new and updated documentation is available to help you on your way to successful Mac OS X application and peripheral development at:

<http://developer.apple.com/techpubs/>

- **Q&A QA1000 - Gathering system information under Traditional Mac OS**

- SAMPLECODE - Graphics 3D: Carbon SetupGL
- SAMPLECODE - Graphics 3D: OpenGL Movie
- SAMPLECODE - Platforms and Tools: Project Builder: Cocoa\_With\_Carbon\_or\_C++
- SAMPLECODE - Networking: OTStreamLogViewer
- SAMPLECODE - Networking: OTClassicContext
- SAMPLECODE - Devices and Hardware: Drivers: TradDriverLoaderLib

### Mac OS X: An Overview for Developers

This 10-page PDF highlights the benefits of Mac OS X technologies to developers.

<http://developer.apple.com/macosex/>

### Fatbrain's Apple Developer Library

The following four titles have been added to the Apple Developer Library, the ADC's print-on-demand series of titles located at:

<http://www1.fatbrain.com/apple/>

- Carbon Porting Guide for Mac OS X
- Inside WebObjects: WebObjects Overview
- Inside Mac OS X: Kernel Environment
- Inside Cocoa: Object-Oriented Programming and the Objective-C Language

## Upcoming Seminars and Events

For more information on Apple developer events please visit the developer Events page at:

<http://developer.apple.com/events/>

### Training and Seminars

#### *Programming with Cocoa*

Taught by Aaron Hillegass at the Big Nerd Ranch, Ashville, NC and Atlanta, GA. Five-day classes are taught on developing web-based and Mac OS X applications.

<http://www.bignerdranch.com/when.html>

### Developer Related Conferences

#### *Game Developer Conference, San Jose, CA*

March 20-24

For 15 years, the Game Developers Conference (GDC) has been the independent and unbiased forum where game development professionals from around the world gather to share ideas and build the skills essential to creating the next generation of interactive entertainment. Apple will be exhibiting at this year's conference.

#### *Worldwide Developers Conference (WWDC) 2001, San Jose, CA*

May 21-25

Register now for Apple's Worldwide Developers Conference 2001, which takes place in San Jose, California from May 21-25. ADC Premier members receive a free pass to the conference, ADC Select members receive discounts for early registration, and ADC Student members can apply for a scholarship to attend for free! For schedules and other details check out:

<http://www.apple.com/developer/wwdc2001/>

If you missed last year's conference, a great way to get ready for WWDC 2001 is to purchase our five volume WWDC 2000 Conference Sessions on DVD-ROM. This set contains over 80 hours of detailed technical information from WWDC 2000 and is deeply discounted for ADC Members. Additional information on this product can be found at:

<http://developer.apple.com/products/>

#### *MacHack Conference, Dearborn, MI*

June 21-23

MacHack, in its sixteenth year, remains centered around cutting edge software development. MacHack's uniqueness derives from the informal feel and the LIVE coding that occurs around-the-clock during the conference.

<http://www.machack.com/>

## Carbon: An Essential Element of Mac OS X

*Continued from page 59*

### An Investment in the Future

Carbon is more than just a porting tool. It's a full-featured API that provides a solid foundation for serious ongoing development. Apple is working hard to make sure Carbon has everything you need to write world-class applications, now and in the future.

Mac OS X, with its killer graphics and superior performance, presents an opportunity for you to revitalize or even reinvent your application. Your goal should be more than simply getting your code running on Carbon—it should be to create a compelling new product that takes advantage of everything Mac OS X has to offer.

When you're finished, be sure to let Apple know about your product, so we can list it on the "Built for Mac OS X" web page located in the Mac Products Guide (MPG):

<http://submit.macsoftware.apple.com/usupdate/upindex.html>

### Until Next Time

With Mac OS X shipping on March 24, the demand for native applications will be huge. Carbon is one way to bring your Mac OS X product to market quickly. Future articles will cover Cocoa, Java, and other options for Mac OS X development.

Whatever path you choose, Apple's Worldwide Developer Relations team is here to help. For technical support, visit <http://developer.apple.com/products/techsupport/>. Better yet, sign yourself up on the Carbon Development discussion list ([carbon-development@lists.apple.com](mailto:carbon-development@lists.apple.com)). This list is monitored by Apple technology managers, engineers, and your fellow developers from around the world. It's a great place to get answers or share your knowledge, and it's free. To sign up, go to <http://lists.apple.com> and search for Carbon Development. See you there!

## Discounts on Mac OS X Migration

### Bring Your Apps to Mac OS X

With Mac OS X Public Beta here, and the commercial release just over the horizon, Mac developers everywhere have a huge need for Carbon and Cocoa application porting, Aqua user interface implementation, and driver development services. Several high-quality software engineering firms, in association with the Apple Developer Connection, are offering these services at very attractive discounts to all ADC Select and Premier members.

<http://developer.apple.com/mkt/macosexmigration.html>

### End-User Tech Support

We all know how expensive and time consuming it is to set-up and maintain a comprehensive customer tech support department. To support developers in their migration to Mac OS X, 877MacTank has designed an exclusive and affordable tech support staffing program for ADC Premier and Select members.

<http://developer.apple.com/mkt/mactank.html>

## Did You Know?

**M**uch of Apple's technical documentation is geared toward helping Carbon application developers. To see the scope of this documentation, check out the Carbon documentation suite on the Web at:

<http://developer.apple.com/techpubs/macosx/Carbon/carbon.html>

Here you will find the "Inside Carbon" suite of API documentation for the nearly 100 Carbon-compatible managers, organized by topic. This documentation includes Carbon inclusion status, per-function release availability, and porting notes to help you modify your legacy Mac OS code for functions that aren't supported. Some documents also describe concepts and include sample code to help you revise your legacy Mac OS applications for Carbon and to create new Carbon applications. You'll also find at the Carbon documentation site a number of relevant programming guides, such as "Aqua Human Interface Guidelines," "Carbon Porting Guide," and "Moon Travel Tutorial: Creating a Carbon Application."

The Carbon documentation suite is also available on the Mac OS X Developer CD. When you install the Developer package, you can also access this documentation from the Developer Help Center using the Help Viewer application.

If you have legacy Macintosh applications that you wish to port, be sure to use the Carbon Dater tool. Carbon Dater produces an HTML report customized for your application. The report lists function calls and code usage that you may need to change, and offers the latest documentation to help you make those changes. Although the Carbon documentation site provides a link to Carbon Dater, the tool itself is not included on the Developer CD.

## Carbon Tips & Tricks

The Apple Developer Connection web site titled Carbon "Tips & Tricks" is dedicated to sharing special techniques for debugging and building Carbon applications.

Visit us at:

<http://developer.apple.com/macosx/carbon/tipsandtricks.html>

By Ron Davis

# Putting An MP3 Player In The Dashboard

## Adding Code To Play MP3 Files Into The PowerPlant Dashboard Example

### INTRODUCTION

To me, one of the coolest things in programming is when something that someone else wrote works easily. You make one call and something you know took a lot of doing just happens, as advertised. Unfortunately this is a rare thing in the world of the programmer, but in this article you'll see one of those things happen, thanks to the QuickTime team. We'll see how to write an application that plays MP3 files, with an absurdly small amount of code.

One caveat before we start. There is more than one way to play an MP3 file from code on the Mac; this is just one. I recently read a comment in an article in the Game Developer Magazine that is relevant here, "Fast, cheap, good — pick two." This article will show you how to do it fast and cheap. My personal opinion is that in most cases it's still pretty good, but there is a problem with QuickTime playback on cooperative operating systems that causes skipping in some cases. The implementation this article uses will skip, especially when it is in the background and you are doing other things in the front. The good news is that it won't skip under Mac OS X, and we'll be writing a Carbon application.

### STARTING WITH THE POWERPLANT DASHBOARD

We're going to use Metrowerk's application framework PowerPlant to implement this application. Why PowerPlant? Because I like it and it gives us a starting place. The starting place is the Appearance Stationary. Create a new Project. Click the "Mac OS PowerPlant Stationary" wizard. Tell CodeWarrior where you want to save. When the Stationary window comes up, expand Carbon, and select Appearance. CW will then create a folder on your drive with the Dashboard application in it. Dashboard is a simple application with one window. It is this window we are going to turn into our MP3 player's controller.

Let's start out in Constructor and make our controls. Open the AppResources.ppob file. Double click "Appearance Window". Delete all of the objects in the window. Now add 3 LCmdBevelButtons, "Play", "Stop", "Pause". We're going to use

LCmdBevelButtons so we don't have to add listening code to our class. Instead, we can just hook into the Command handling mechanism already in place. When configuring each button, you need to change three fields: the Pane ID, Title, and Command Number. When you get back to the project, you will need to add constants for these commands. Table 1 shows the values I used.

| Title | Const     | Pane ID | Command Number |
|-------|-----------|---------|----------------|
| Play  | cmd_Play  | 'play'  | 1000           |
| Stop  | cmd_Stop  | 'stop'  | 1010           |
| Pause | cmd_Pause | 'paus'  | 1020           |

Table 1. Constant and command values.

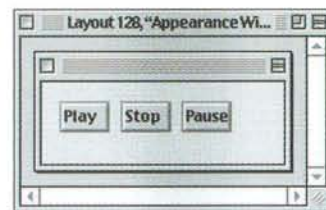


Figure 1. Controller Window in Constructor.

In Constructor, the controller window looks like **Figure 1**. Since we used LCmdBevelButtons, we have to add menu items with the command numbers. Add a new menu in Constructor. Call the menu "Controls." Give it three commands with titles and command numbers just like the buttons. Don't forget to add the menu to the Menu Bar (or it won't show up in your application).

At this point, you can run the application and a window will show up with all of the buttons disabled.

### WRITING THE CODE

Now we need to write some code to make our player work. Everything is going to happen in the CAppearanceApp class. We'll handle opening the MP3 file and the Play, Stop and Pause controls from here. In order to do this, we are going to have to add a few routines and instance variables. Open the header for CAppearanceApp. We need to add a variable that holds the FSSpec of the currently playing

**Ron Davis** is a consulting Engineer in Apple's iServices group. He's also the author of his own more complete MP3 player, MP3 Hit List, which is soon to be released.

- Custom Destinations

- Electronic Transaction Processing

- Easy Trialware Creation

- Install or UnInstall

# Get It Together With InstallerMaker™

- Installation From FTP or HTTP Sites

## *When Time Is Money, Get It Done Fast!*

**Build Smaller Installers** – With StuffIt InstallerMaker, you'll build installers faster than ever before! Compress your installers an average of 15% smaller using the power of the StuffIt Engine®. Smaller installers download faster, provide added space on CDs and servers, and increase network bandwidth.

**Quickly Create Demoware** – Easily turn your application into a polished demo. Just set the number of days for the demo to be active, paste in the graphics, and you're done!

**Archive Freshening** – Automatically update your installer project file; eliminate repeated searches for modified files.

- Resource Installation

- Built-In Resource Compression

- ShrinkWrap Disk Image Support

**Scripting Saves Time** – InstallerMaker provides full scriptability for all features. Automating the build process saves time and helps ensure the integrity of your installer.

**Trialware in Minutes** – Creating trialware is a breeze with InstallerMaker. With just a few clicks, and no extra coding, you can create trialware that is e-commerce ready.

**Update in a Flash** – Easily build intelligent "diff" files in installers to create small updaters for quick online distribution. From one simple installer, you can update 68k, PowerPC or FAT applications.

**More Details Online** – There's a lot more InstallerMaker can do for you. Get all the info at [www.aladdinsys.com](http://www.aladdinsys.com).

- Marketing Opportunities To Help You Make Money

- Hierarchical Package Support

- Built-In Updaters



MP3 file and a QuickTime **Movie** variable of the currently playing MP3. The standard appearance application doesn't keep track of the window once it creates it, but we need to make sure there is only one window open, since we can only play one MP3 at a time. For this reason, we need to add an instance variable to hold the window. This "one MP3 at a time" limitation is ours, not QuickTime's. You could play multiple sound files at once if you wanted.

QuickTime treats an MP3 just like a movie. As a matter of fact, you could use a movie controller to play the MP3 and you wouldn't need our command buttons. On the other hand, you would lose some level of control and you would have to display something on the screen. Our method will allow us to play an MP3 with only the UI we want, or no UI at all.

There are two other things that need to be added to the class declaration of **CAppearanceApp**. We'll be writing two new methods, **ChooseFile** and **SetUpMP3**. Listing 1 shows the complete **CAppearanceApp** class.

#### Listing 1: Declaring our custom class

CAppearanceApp.h

```
class CAppearanceApp : public LApplication {
public:
    CAppearanceApp();
    ~CAppearanceApp();

    virtual Boolean ObeyCommand(
        CommandT      inCommand,
        void*          ioParam = nil);

    virtual void      FindCommandStatus(
        CommandT      inCommand,
        Boolean&       outEnabled,
        Boolean&       outUsesMark,
        UInt16&       outMark,
        Str255        outName);

    virtual void      ChooseFile();
    virtual void      SetUpMP3( bool startPlaying );
protected:
    virtual void      StartUp();
```

```
void      RegisterClasses();
```

```
LWindow*  mWindow;
FSSpec    mMP3File;
Movie     mMovie;
```

```
};
```

Now let's move into the actual code of the class. We will be starting with the constructor. We need to do a couple of things in here to set up QuickTime. Listing 2 contains the altered constructor.

#### Listing 2: The updated constructor for CAppearanceApp

CAppearanceApp.cp

```
CAppearanceApp::CAppearanceApp()
:mWindow(NULL), mMovie(NULL)
{
    bool cantRun = false;

    // Register ourselves with the Appearance Manager
    if ( UEnvironment::HasFeature(env_HasAppearance) ) {
        ::RegisterAppearanceClient();
    }

    long result;
    if ( ::Gestalt(gestaltQuickTime, &result) != noErr ) {
        // put a dialog here that says we need QT
        cantRun = true;
    } else {
        ::Gestalt(gestaltQuickTimeVersion, &result);
        if ( result < 0x0400 )
        {
            cantRun = true;
        }
    }

    if ( cantRun )
    {
        // you should put up an alert here telling the user why.
        ExitToShell();
    }

    RegisterClasses();
    EnterMovies();
}
```

Intelligent Web site Monitoring and Analysis Software

**FUNNELWEB**  
**ENTERPRISE**

now for the Mac OSX

Any Server, Any platform, Any time

[www.funnelweb4.com](http://www.funnelweb4.com)

[info@activeconcepts.com](mailto:info@activeconcepts.com)

Toll free 1800 551 5282

Speed, intuitive user interface, accuracy and in-depth analysis have always made Funnel Web the intelligent choice for Web site analysis.

Now includes pdf output, incremental analysis, cluster analysis and streaming media reports.

**Visit our booth #3240 in the MacTech Pavillion and save 40% when you purchase Funnel Web Enterprise. Show Special for one week only.**

# "I registered my practice"

[www.thecattherapist.com](http://www.thecattherapist.com)



Carole Wilbourn, world renown Cat Therapist, registered her domain name to put her business online at [register.com](http://register.com). Visit us at [www.register.com](http://www.register.com) or call us at 1-800-7-WWW-NET, and we'll help you register your domain name right over the phone.

**register**  
**com**<sup>TM</sup>  
the *first* step on the web<sup>SM</sup>

The first thing we do is initialize our `mWindow` and `mMovie` variables to `NULL`. We are going to check them later to determine if the window or movie has already been created, so they had better be `NULL` the first time we check it.

We need to check for QuickTime. We do this with the `Gestalt` calls in the center. I went ahead and checked for QuickTime version as well. If we don't have QuickTime, we exit the program right here. If we do have it, we register our classes and call `EnterMovies`. `EnterMovies` is the QuickTime initialization call and must be made before calling any other QuickTime Routines.

Listing 3 contains the new `CAppearanceApp::Startup` routine. The default behavior of Appearance is to open a new window at start up. But what does New mean in our app? This isn't a recording app; there are no new MP3s to be created. So what we are going to do is stop New from happening at all. While we're at it, we'll make Open happen in its place, which will let us open an MP3 file and play it.

### Listing 3: The start-up routine

```
void CAppearanceApp::Startup()
{
    ObeyCommand(cmd_Open, nil);
}
```

Now we need to make sure the New menu command isn't even available to the user and that the commands for our controls are enabled. Go to `FindCommandStatus` and change the `cmd_new` to `cmd_open`. Then add cases for the constants we defined for our commands. When it is done, your `FindCommandStatus` should look like Listing 4.

### Listing 4: Adjusting our menu items

```
CAppearanceApp::FindCommandStatus

void CAppearanceApp::FindCommandStatus(
    CommandT inCommand,
    Boolean& outEnabled,
    Boolean& outUsesMark,
    UInt16& outMark,
    Str255 outName)
{
    switch (inCommand) {
        case cmd_Open:
        case cmd_Play:
        case cmd_Stop:
        case cmd_Pause:
            {
                outEnabled = true;
                break;
            }
        default: {
            LApplication::FindCommandStatus(inCommand, outEnabled,
                outUsesMark, outMark, outName);
            break;
        }
    }
}
```

To handle the Open menu command, we need to change a couple of things. As all PowerPlant programmers know, you do the actual work of a menu command in the `ObeyCommand` method and that is where the code is right now to display the window. We're going to change that a little. We need to pick a file before we open the window, so we're going to create a new method called `ChooseFile` to handle picking the file and opening the window. Listing 5 contains the complete `ChooseFile` method.

### Listing 5: Choosing a file

```
CAppearanceApp::ChooseFile

void CAppearanceApp::ChooseFile()
{
    PP_StandardDialogs::LFileChooser chooser;
    NavDialogOptions* options = chooser.GetDialogOptions();
    if (options != nil) {
        options->dialogOptionFlags = kNavDefaultNavDlogOptions
            + kNavNoTypePopUp
            + kNavAllowMultipleFiles;
    }
    chooser.SetObjectFilterProc( NavServFileFilterProc );
    if (chooser.AskOpenFile( LFileTypeList(fileTypes_All))) {
        // let the filter proc handle types
        chooser.GetFileSpec( 1, mMP3File);
        if ( mWindow == NULL )
        {
            mWindow = LWindow::CreateWindow(PPob_SampleWindow, this);
            ThrowIfNil_(mWindow);
            mWindow->Show();
        }
        SetUpMP3( true );
    }
}
```

A number of things are happening in this routine. We are using PowerPlant's `LFileChooser` to handle the open dialog. Under Carbon, this will always use Navigation Services. First we set up Navigation Services options, telling it not to use the types pop-up menu and to only pick one file. (Yes, adding `kNavAllowMultipleFiles` only lets you pick one file. Go figure.) Then we tell the `LFileChooser` and Nav Services to use our filter procedure. We'll talk about the filter proc in a minute. After the set up, we do the `AskOpenFile`, which will be handled by putting the dialog up and getting its results. If the user doesn't cancel, it will return true and we will execute the code inside our if.

The first thing we do is get the file system specification of the file

# USBStuff

1-88-USB USB US  
-or- (1-888-728-7287)

WorldWide Distributors of USB  
and FireWire Parts,  
Peripherals and Accessories

Hey Developers:

<http://www.usbstuff.com/developers.html>

1-877-4 HOTWIRE  
-or- 1-877-446-8947

# FireWireStuff

the user picked. We assign it to our instance variable for later use. Then we create the window if it doesn't exist. Once the window exists, we call `SetUpMP3`, which we are going to write next.

Before we go on, Listing 6 shows the Nav Services Filter proc I'm using. It checks to see if the type of the file is one of the two MP3 file types I know about and it also checks to see if the file ends with `.mp3` or `.MP3`, which should catch most MP3 files.

#### Listing 6: Filter files for the Open dialog box

```

NavServFileFilterProc
pascal Boolean NavServFileFilterProc (
    AEDesc* theItemAEDesc, void* info,
    NavCallBackUserData /*callBackUD*/,
    NavFilterModes /*filterMode*/)
{
    OSErr theErr = noErr;
    Boolean display = false;
    NavFileOrFolderInfo* theInfo = (NavFileOrFolderInfo*)info;
    if ( theItemAEDesc->descriptorType == typeFSS )
    {
        if ( theInfo->isFolder ) // show folders and volumes
        {
            return true;
        }
    }
    StAEDescriptor aSpecDesc;
    if (::AECoeerceDesc(theItemAEDesc, typeFSS, aSpecDesc) == noErr) {
        FSSpec spec;
        OSErr err = ::AEGetDescData(aSpecDesc, &spec,
            sizeof(FSSpec));
        ThrowIfOSErr_(err);
        {
            FInfo finderInfo;
            ::FSpGetFInfo( &spec, &finderInfo );
            LStr255 tempString( spec.name );
            if tempString.EndsWith( "\p.mp3" ) ||
                tempString.EndsWith( "\p.MP3" ) )
            {
                display = true;
            }
            if ( finderInfo.fdType == 'MPG3' ||
                finderInfo.fdType == 'MP3 ' )
            {
                display = true;
            }
        }
    }
    return display;
}

```

#### LOADING AND PLAYING THE MP3

Now we're ready to load the file and play it. The loading of the file takes place in the method `SetUpMP3`. Listing 6 contains the complete routine. Playing an MP3 using QuickTime is just like playing a movie. The only difference is we don't have to worry much about where it draws.

```

void CappearanceApp::SetUpMP3( bool startPlaying )
{
    // mMP3File has been set with the file we want to play.
    // set it up to play.
    if ( mMovie != NULL )
    {
        DisposeMovie(mMovie);
        mMovie = NULL;
    }
    mWindow->SetDescriptor(mMP3File.name);
    // get the movie from the file
    OSErr err;
    SInt16 movieRefNum;
    err = ::OpenMovieFile(&mMP3File, &movieRefNum, fsRdPerm);
    ThrowIfOSErr_(err);
    SInt16 actualResID = DoTheRightThing;
    Boolean wasChanged;
    err = ::NewMovieFromFile(&mMovie, movieRefNum, &actualResID,
        nil, newMovieActive, &wasChanged);
}

```

# Mac OS X

## Porting & Development Showcase

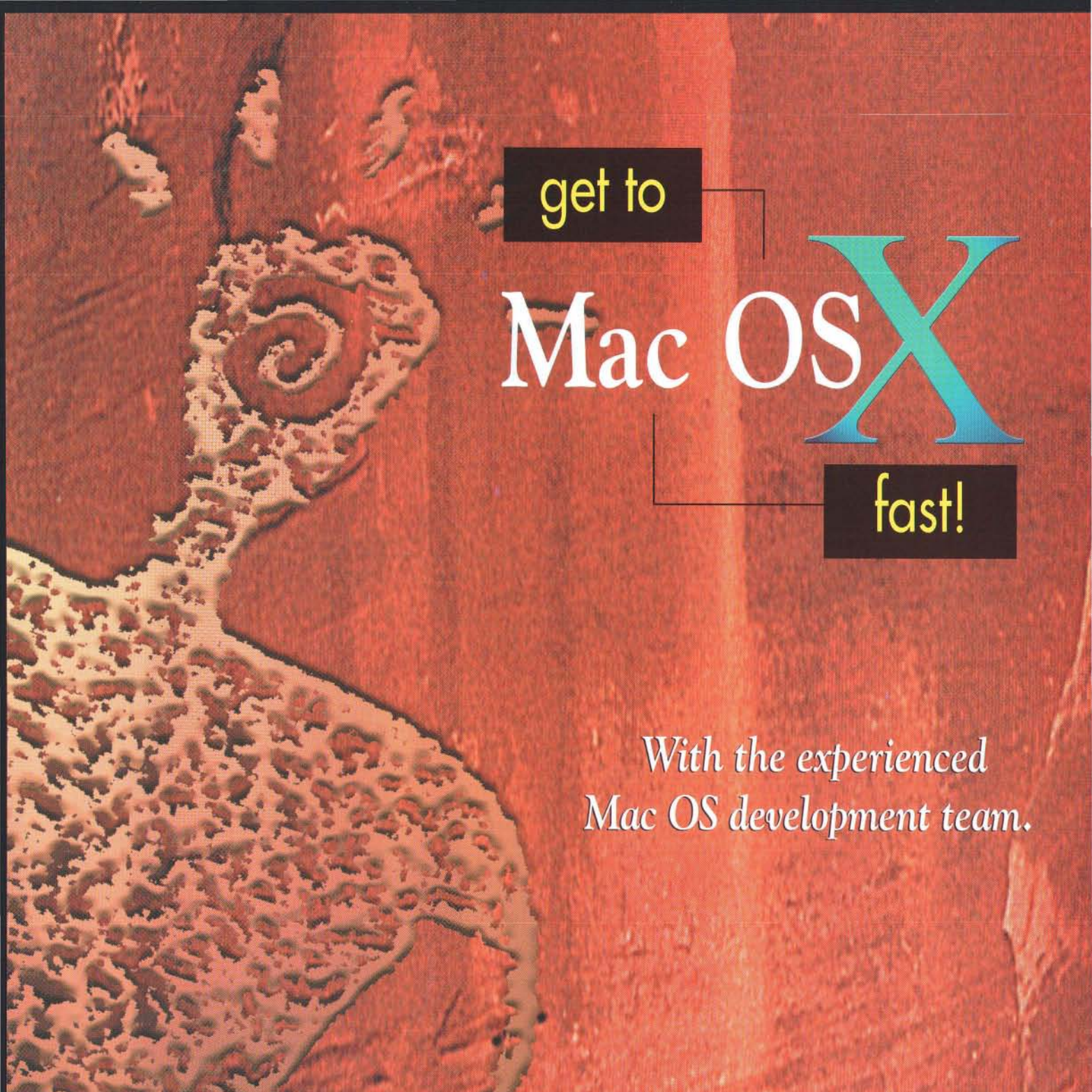
# Mac OS X

## Making the Move

With **MAC OS X** just over the **HORIZON**,  
 Mac *developers* everywhere have a  
 major need for **CARBON** and **COCOA**  
 application porting, **AQUA** user interface  
 implementation, and **DRIVER** development  
 services. Toward that end, several  
 high-quality **SOFTWARE** engineering  
 firms, in association with the **APPLE**  
**DEVELOPER CONNECTION**, are offering these  
 services at very *attractive* discounts  
 to all **ADC** Select and Premier members.

The following pages  
 are those vendors that  
 are part of the Mac OS X

Porting & Development Showcase.



get to

Mac OS X

fast!

*With the experienced  
Mac OS development team.*

Red Rock Software specializes in Macintosh software development. Our team of senior engineers averages over 10 years of development experience on the Macintosh platform. Our experience and expertise has gained us the trust of companies like *Apple Computer, Iomega Corporation and Sorenson Media.*

If you are looking to make your product compatible with Mac OS X quickly, let our experts help.

- Aqua Interface Implementation
- OS X Carbon Porting
- OS X Native Cocoa Application Development

R E D R O C K

*s o f t w a r e*

Call Red Rock at **888.689.3038**  
or visit us online  
at **[www.redrocksw.com](http://www.redrocksw.com)**

Programming at the Speed of Light

Aqua

Cross-Platform

Device Drivers

TCP / IP

Graphics

Plug-Ins

Mac/Unix/Win32

Ports

Carbon / OS X



Software Development  
Outsourcing Since 1990

415.491.2200  
[www.shadetreeinc.com](http://www.shadetreeinc.com)

- ☐ Eat your vegetables.
- ☐ Exercise every day.
- ☒ Port to Mac OS X.
- ☐ Call your mom.

All of these are good for you.  
We can make one of them easy.

Since 1989, The Omni Group has worked with the technologies that have been refined into Mac OS X.

Moving to Mac OS X is a big step for your company, and you need consultants who can help you both plan how best to make the transition and follow whatever path you choose.

That's been our business for years. No matter what kind of product you have, we can get it up under OS X, fast:

- Real games: We ported id's Doom and Quake games to NEXTSTEP and OS X. Quake 2 took us a week.
- Big apps: We ported Adobe's FrameMaker to NEXTSTEP and Sun's Concurrency to OpenStep/Solaris.
- Big libraries: We ported the Oracle 8 client libraries which Apple ships today in OS X Server.
- Serious drivers: We ported 3dfx's Glide and wrote Voodoo2 and Rendition drivers for OS X Server.  
We've written new mouse drivers for OS X Server and joystick drivers for OpenStep.
- New apps: We wrote OmniWeb, the only native OS X web browser, and OmniPDF, the native Acrobat viewer for OS X.

Mac OS X is what we do. Let us help you do it, too.

## THE OMNI GROUP



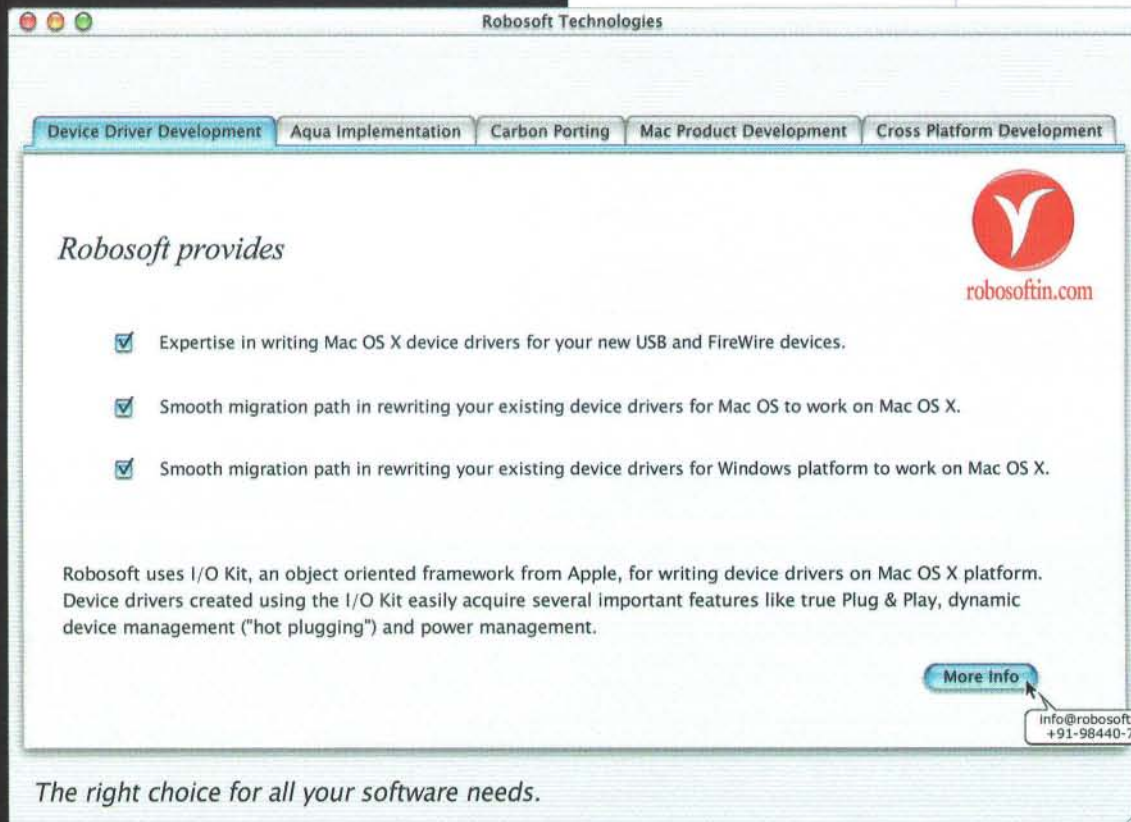
2707 Northeast Blakeley Street  
Seattle, Washington 98105-3118  
[www.omnigroup.com/consulting](http://www.omnigroup.com/consulting)

[sales@omnigroup.com](mailto:sales@omnigroup.com)  
800.315.OMNI x201  
206.523.4152 x201

# Device Driver Development?

Special  
Discounts  
for  
ADC Members

Get the most **experienced**  
**Mac product development** company  
working for **you.**



Robosoft Technologies

Device Driver Development Aqua Implementation Carbon Porting Mac Product Development Cross Platform Development

*Robosoft provides*

- ✓ Expertise in writing Mac OS X device drivers for your new USB and FireWire devices.
- ✓ Smooth migration path in rewriting your existing device drivers for Mac OS to work on Mac OS X.
- ✓ Smooth migration path in rewriting your existing device drivers for Windows platform to work on Mac OS X.

Robosoft uses I/O Kit, an object oriented framework from Apple, for writing device drivers on Mac OS X platform. Device drivers created using the I/O Kit easily acquire several important features like true Plug & Play, dynamic device management ("hot plugging") and power management.

[More info](#)

[info@robosoftin.com](mailto:info@robosoftin.com)  
+91-98440-77232

*The right choice for all your software needs.*

## Also contact us for:

- ✓ Porting your existing Windows products to the Macintosh Platform.
- ✓ Porting (Carbonizing) your existing Mac OS products to Mac OS X.
- ✓ Providing integrated eBusiness solutions using WebObjects.
- ✓ Designing and Developing Cross Platform products.

**Partners  
in Product  
Development**

```

ThrowIfOSErr(err);
err = ::CloseMovieFile(movieRefNum);
ThrowIfOSErr(err);
// start the file playing
Rect movieBox;
::GetMovieBox(mMovie, &movieBox);
::OffsetRect(&movieBox, -movieBox.left, -movieBox.top);
::SetMovieBox(mMovie, &movieBox);
::SetMovieGWorld(mMovie, mWindow->GetMacPort(), nil);
::PrerollMovie(mMovie, 0, ::GetMoviePreferredRate(mMovie))
// tried asking for less than the whole movie but it seems to load it all anyway.
err = ::LoadMovieIntoRam(mMovie, 0,
::GetMovieDuration(mMovie), unkeepInRam);
if (startPlaying)
{
    ObeyCommand(cmd_Play);
}
}

```

*Listing 6: SetUpMP3, the routine that loads our MP3.*

Our program is only going to play one MP3 at a time, and we will reference that MP3 internally as a QuickTime Movie type. Our reference variable is `mMovie`, which we initialized to `NULL` in the constructor. The first thing we do in our routine is check to see if there is already an existing `mMovie`. If so, we need to dispose of it properly via QuickTime's `DisposeMovie` routine.

Now we have a clean slate and the first thing we do is set the title of our window to the name of the file we are going to play. Then we load the movie. We tell QuickTime to `OpenMovieFile`. If there is a problem opening the file, like it isn't really an MP3, then we throw and get out of the routine.

Next we tell QuickTime to create a new Movie for the file, `NewMovieFromFile`. This routine takes our `mMovie` variable, the `movieRefNum` we got from our open call, the actual `ResID`, flags that tell it we want the movie active, and a flag telling us if they had to change any references we gave them. Notice we set `actualResID` to the name of a Spike Lee movie. This variable is the resource ID of the first resource in our movie file. We set it to `DoTheRightThing`, in a vain attempt to get Spike's movie, but we won't and QuickTime will set our variable to the actual resource ID, or in our MP3 case, nothing.

Now we have got the movie in QuickTime and we can close the file. Before we can play the file we need to set up its "Movie Box," which is where we want it to play. We do this by getting the current box, making it nothing and assigning the `GWorld` to our window. None of this really matters to us, but it does to QuickTime.

We do a couple of things to try and minimize skipping. First we `preroll` the movie. This causes the beginning of the movie to be buffered. If you were opening a stream, you'd have to call `PrePrerollMovie`. Then we try and load all of the movie into RAM. If this succeeds we'll eliminate skipping. The last parameter of the call is set to `unkeepInRam`, which is a constant that will allow the memory to be purged if it is needed. You can set it to `keepInRam`, which won't allow the memory to be freed. I did this and it sucked up a huge amount of RAM and brought my MP3 player to a screeching halt. Remember most MP3s are rather big. I've got the Rush 2112 track I ripped, which is 21 minutes long and 19 Meg on disk. So loading isn't always practical. Of course, given OS X's different memory model, loading probably works well.

Before we leave the routine, we check its parameter and if it is true we give the command to play, which will start the file playing.

The last routine we need to write is the `ObeyCommand` method.

This is a standard PowerPlant method that handles menu commands and will handle our buttons. It is in Listing 7.

```

Boolean CAppearanceApp::ObeyCommand(
    CommandT inCommand,
    void* ioParam)
{
    Boolean cmdHandled = true; // Assume we'll handle the
    command
    switch (inCommand) {
        case cmd_Open:
        {
            ChooseFile();
        }
        break;
        case cmd_Play:
        {
            ::StartMovie(mMovie);
        }
        break;
        case cmd_Stop:
        {
            ::StopMovie(mMovie);
            ::SetMovieTimeValue(mMovie, 0);
        }
        break;
        case cmd_Pause:
        {
            ::StopMovie(mMovie);
        }
        break;
        default: {
            cmdHandled = LApplication::ObeyCommand(inCommand,
            ioParam);
            break;
        }
    }
    return cmdHandled;
}

```

*Listing 7: Obeying commands for the player.*

The first command we handle is the open command, which just calls our `ChooseFile()` method. Next we handle the Play command. To start our already loaded movie playing, we just call QuickTime's `StartMovie` with our `mMovie` as the parameter. To Pause the movie, we call `StopMovie`. `StopMovie` doesn't move where the "playhead" is for the movie, so if you call `StartMovie` again it will start playing where it was. The proper behavior for a stop button is to go back to the beginning of the song. So to handle the Stop command, we stop the movie and then move the play position back to the beginning, using `SetMovieTimeValue`. You could use `SetMovieTimeValue` to set the play head wherever you wanted in the song. You could even hook it up to a slider and let the user do it. I'll leave that as an exercise for the readers.

## CONCLUSION

That is the basics of playing an MP3 file using QuickTime. If you want to learn more you need to check out the QuickTime API web site, and maybe subscribe to the QuickTime API mailing list.

## BIBLIOGRAPHY

- QuickTime API:  
<<http://developer.apple.com/techpubs/quicktime/qtdevdocs/RM/frameset.htm>>
- QuickTime API Mailing List:  
<<http://lists.apple.com/mailman/listinfo/quicktime-api>>
- 2112, Rush
- Do The Right Thing, Director, Spike Lee.

MT

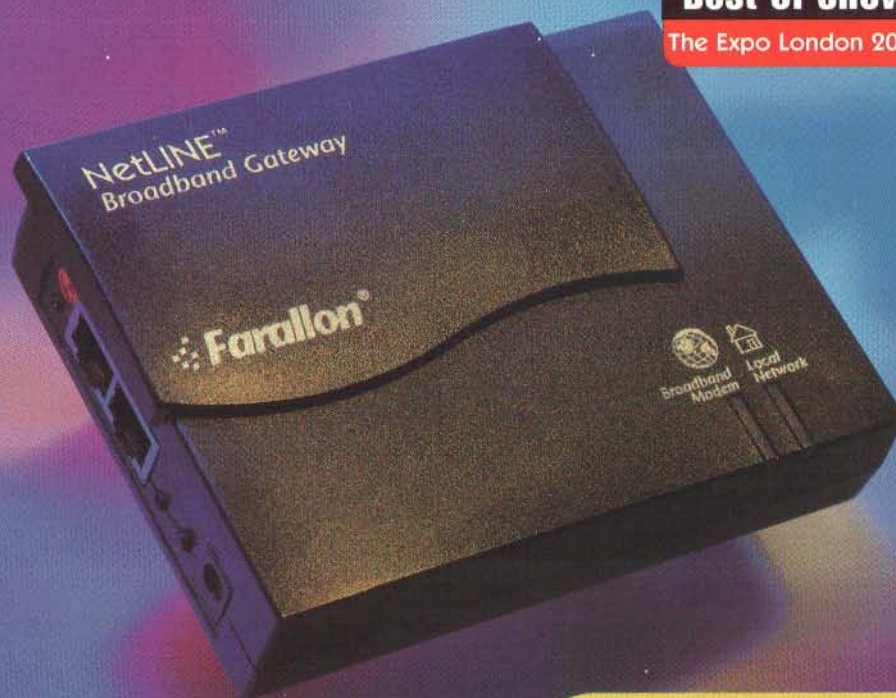
Share your high-speed Internet connection and protect your data from unauthorized access with the

# NetLINE™ Broadband Gateway

**Farallon**  
www.farallon.com

**Best of Show**

The Expo London 2000



*The perfect solution for home, office or school!*

## The NetLINE Broadband Gateway

allows you to share a high-speed Internet connection, such as Cable or DSL, with multiple computers using just a single IP address from your ISP.

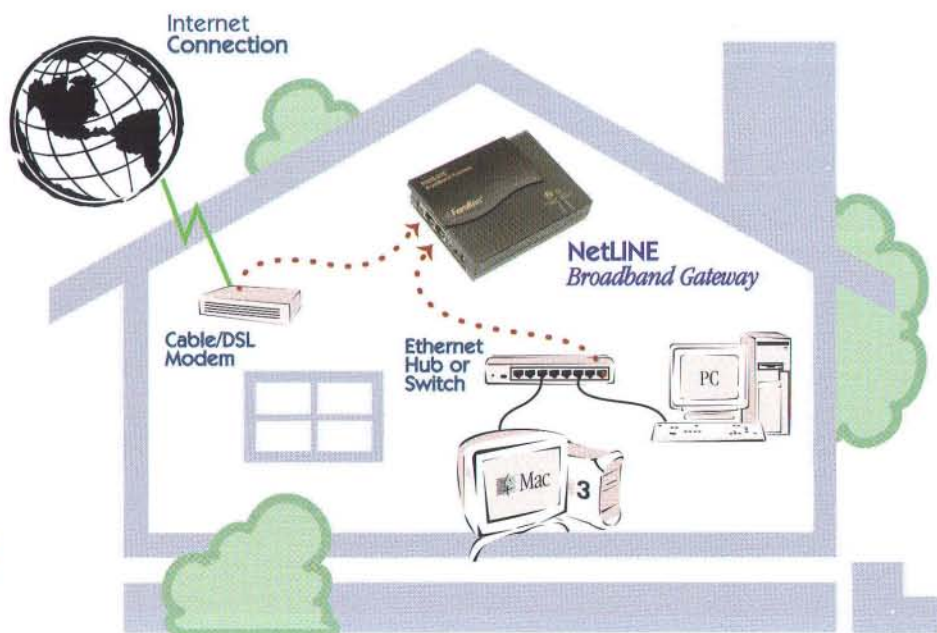
The built-in firewall will protect your important files by preventing unauthorized access via the Internet.

Works with both Macs and PCs. Web-based configuration makes set-up a breeze.

Receive a coupon for 25% off a Farallon hub or switch when you purchase the NetLINE Broadband Gateway. (US Only)



For more information contact Dr. Farallon at 1-800-613-4954 or visit us on the web at [www.farallon.com](http://www.farallon.com)



The NetLINE Broadband Gateway connects between your network and your Cable/DSL modem

Available at **DEPOT**

by Jeff Clites <online@mactech.com>

### SMALLTALK

I recently ran across a paper which describes a study comparing the productivity and efficiency of several programming languages. While it's difficult to draw concrete conclusions from such a comparison, the paper did make an interesting observation: programmers of scripting languages (such as Perl, Python, and Tcl) tend to use language features differently than programmers of non-scripting languages (such as C, C++, and Java). Specifically, they noted that the former group used data structures (such as arrays and dictionaries) which are built into the language, where as the latter group consistently implemented their own data structures from scratch rather than using those provided by their languages' standard libraries. This reminded me of an observation that I had made of my own programming practices — namely, that learning a new programming language influences how I program in *other* languages. For example, after learning Perl I tended to use associative arrays (also known as hashes, dictionaries or maps) more frequently in other languages, and implemented custom objects and data structures less often. In fact, this makes a lot of sense when you start to think about it — different languages have different design goals and cultures, and lead you to thinking about programming from different perspectives. As a developer, I view a programming language as a tool — albeit a rather important one. Like most other tools, people tend to have a favorite, but you are always better off having a variety to call upon to tackle different problems. And with programming languages in particular, I would argue that knowing a variety of languages can make you a better programmer in your language of choice.

So in the spirit of broadening our horizons, this month we are going to talk about the programming language Smalltalk. The goal here isn't to convince you to use Smalltalk for your next project, but rather to encourage you to read up on, and play with, a new language, as an incubator for new ideas and because, frankly, it's fun.

An empirical comparison of C, C++, Java, Perl, Python, Rexx, and Tcl  
<[http://www.wipd.ira.uka.de/~prechelt/Biblio/jccprt\\_computer2000.pdf](http://www.wipd.ira.uka.de/~prechelt/Biblio/jccprt_computer2000.pdf)>

### WHAT MAKES SMALLTALK SPECIAL

Smalltalk has been around for quite a while, and in fact its genesis dates back to Alan Kay's Dynabook project at Xerox PARC, the same project which lead to the development of the graphical user interface (GUI). It was one of the things shown to Steve Jobs on his famous trip

to the research center, and according to the story Xerox was more worried about letting Smalltalk out of the bag than they were about revealing the GUI.

Smalltalk was arguably the first pure object-oriented language; everything in Smalltalk is an object — there are no primitive types. It will look a bit familiar to Objective-C programmers — the message-passing syntax (as well as the object model) of Objective-C was based on Smalltalk, the most obvious difference being that Smalltalk does not enclose its message sends in brackets. (So, the Objective-C statement “[dog fetch:stick]” would be just “dog fetch:stick” in Smalltalk.) Smalltalk was a language ahead of its time in many ways. For instance, inherent to all Smalltalk development systems is a class browser. You've probably seen class browsers before, either in connection with Java, or in Apple's Project Builder, or in some other IDE. What sets the Smalltalk Browser apart is that all development is done there — Smalltalk source code does not live in separate text files. The plus side of this is that the programmer doesn't have to worry about how source code should be organized, or about what classes and methods live in what files — this is taken care of by the development environment, freeing the programmer of all of the mundane details. This approach also simplifies the language: for instance, there is no syntax for declaring a new class, because none is needed — you simply use the “new class” command in the browser GUI. (On the other hand, one of the down sides of this is that it is awkward to print out Smalltalk code, or to include it in email messages.)

Another seemingly strange detail you will notice is that there is no “main” entry point to a Smalltalk program, or even clear distinctions between programs. To run some code, you simply select it in the Browser and choose “do it” from the menu. This melding of the development environment into an integral part of the language may feel uncomfortable to programmers who are used to a strict separation between the two, but if you give it a chance you may find it refreshing. Although Smalltalk is not new, somehow this approach feels like progress over the predominant “programs == text” paradigm.

A must-read for anyone interested in Smalltalk is *Design Principles Behind Smalltalk*, a high-level overview of the philosophy behind the language, originally printed in the August 1981 edition of *Byte Magazine*, which marked the first public appearance of Smalltalk. A brief, no-nonsense description can also be found in the TUNES Project's *Review of Existing Languages*. Also, you'll want to refer to either the Google Web Directory or the Open

Directory pages on Smalltalk for links to numerous other resources, and to both of the major FAQs for answers to many common questions. Finally, the online version of AOKI Atsushi's book *Smalltalk Idioms* is a fair-sized tutorial which has a nice introductory section to set the conceptual stage.

#### Design Principles Behind Smalltalk

<[http://users.ipa.net/~dwighth/smalltalk/byte\\_aug81/design\\_principles\\_behind\\_smalltalk.html](http://users.ipa.net/~dwighth/smalltalk/byte_aug81/design_principles_behind_smalltalk.html)>

#### Review of existing Languages

<<http://www2.tunes.org/Review/Languages.html#Smalltalk>>

#### Google Web Directory - Smalltalk

<<http://directory.google.com/Top/Computers/Programming/Languages/Smalltalk/>>

#### Open Directory: Smalltalk

<<http://dmoz.org/Computers/Programming/Languages/Smalltalk/>>

#### Dave's Smalltalk FAQ

<<http://www.dnsmith.com/SmallFAQ/>>

#### Smalltalk FAQ (v.1.0)

<<http://www.faqs.org/faqs/smalltalk-faq/index.html>>

#### Smalltalk Idioms

<[http://www.sra.co.jp/people/aoki/SmalltalkIdioms/index\\_e.htm](http://www.sra.co.jp/people/aoki/SmalltalkIdioms/index_e.htm)>

### THE SMALLTALK LANGUAGE AND OBJECT-ORIENTED DESIGN

As I said at the beginning of this article, I see Smalltalk in part as a jumping off point to learn about object-oriented programming in general, to be applied to programming in other languages as well. And Smalltalk is the perfect playground for this. It is a pure object-oriented language — there are no primitive types, and almost everything is handled through messages sent to objects. This in itself simplifies the learning of the language, as there is a consistent semantics to all variables, a feature lacked by hybrid languages such as Java, C++, C#, and even Objective-C. (Take a look at the interesting article *Primitive Types Considered Harmful* on IBM's web site for a discussion of this, in the context of Java.) Continuing the same theme, Smalltalk is completely dynamically typed — variables do not have type declarations, so that all type-related information is contained in the objects themselves, and is accessed at run time rather than at compile time. Also, there are only five reserved words (**self**, **super**, **nil**, **true**, and **false**), and minimal syntax. In fact, there are actually no flow-control structures in the language (such as "if", "for", or "while") — flow control is also handled *via* messages to objects. This will seem strange to programmers of other languages, as almost every language contains the same basic branching and looping constructs (think of C, Java, Perl, and Python). Overall, the clean design of the language promotes a focus on the design of the programs you write with it, as there is minimal syntax to get in the way of learning and using the language. Although Smalltalk was originally designed to be usable by children, its simplicity makes it deep and abstract as well.

### Primitive Types Considered Harmful

<<http://www7.software.ibm.com/vad.nsf/Data/Document2712>>

So how *does* Smalltalk handle situations that call for an "if" or a loop? The language has a construct called a block — some other languages, such as Perl, have similar constructs, but they are not typically used in the same manner that blocks are used in Smalltalk. A block is just, well, a block of code (enclosed in square brackets), and it can be passed as a parameter to a method just like other objects. Conceptually, passing a block to a method is like passing a function pointer in other languages, except that rather than defining a function in one place and then passing the function pointer, you just put the code right where you are going to use it. In practice, blocks are used extensively in Smalltalk (more extensively than function pointers are used in other languages). Using this construct, a loop is implemented by sending to a collection a message (usually "do:") with a block as a parameter — in effect, the message asks the collection (such as a list) to execute the block for each object the collection contains (passing each object to the block as an argument). Similarly, conditionals are implemented by sending the "ifTrue:" message to an expression which evaluates to a boolean, again passing a block as a parameter — so the "true" object knows that it should evaluate the block, and the "false" object knows that it shouldn't. The syntax (as well as the idea) takes a bit of getting used to, but in addition to simplifying the semantics of the language, this approach makes it natural for the programmer to define his own looping and conditional constructs, giving the language a great flexibility while maintaining its clarity. Take a look at the Smalltalk Industry Council's *An Intro to Smalltalk* for an example illustrating this syntax, as well as a brief sketch of Smalltalk's history, and UIUC's *VisualWorks: Smalltalk basics* page for a pithy overview of the most important and interesting features of the language.

#### Smalltalk Industry Council - An Intro to Smalltalk

<<http://www.stic.org/news.ez?viewLink=7>>

#### VisualWorks: Smalltalk basics

<<http://wiki.cs.uiuc.edu/VisualWorks/Smalltalk+basics>>

Once you've become familiar with the basics of Smalltalk, you can move on to some of the discussions of object-oriented programming which fit naturally with the language. Topics include the concept of "super" and whether accessor methods are a good idea. On the web, three good places to start are articles on the Bytesmiths, Instantiations, and UIUC sites. You'll also find many interesting articles in the book *Kent Beck's Guide to Better Smalltalk: A Sorted Collection* (ISBN: 0521644372).

#### Bytesmiths' Smalltalk Publications

<<http://www.bytesmiths.com/pubs/>>

## Smalltalk Products - Publications

<<http://www.instantiations.com/sts/pubs.htm>>

## VisualWorks: Conversation topics

<<http://wiki.cs.uiuc.edu/VisualWorks/Conversation+topics>>

## SQUEAK

Because Smalltalk blurs the lines between IDE, source code, and runtime, a Smalltalk development environment is almost like an operating system unto itself — in fact, early Smalltalk implementations ran on the bare hardware, without an operating system in between. Today, there are several Smalltalk implementations available (some commercial and some free), and each varies from the others in the details of the IDE and even the language itself. One of the more interesting environments is Squeak, a free, open-source, cross-platform, evolving Smalltalk. It is the perfect environment to play with as you begin to experiment (which does not imply that it is a “toy” implementation). Squeak actually originated at Apple, with a group of developers who wanted a non-commercial Smalltalk to use in a project they were working on — a group which in fact included Alan Kay, the driving force behind the original Smalltalk. Its development continued after some of the team moved to Disney, and Squeak is now open-source, with an active community behind it; there is even a newly published book, *Squeak: Object-Oriented Design with Multimedia Applications* (ISBN: 0130280283).

### The Squeak Homepage

<<http://squeak.org/>>

### Squeak for Mac OS X

<<http://www.metaobject.com/downloads/Squeak/>>

Part of its success undoubtedly stems from its unique openness — it is in fact open and accessible on a number of levels. First, as mentioned above, it's open-source. Secondly, it is cross platform and highly portable — one if the features of Smalltalk (and Squeak in particular) is that it defines its own notion of a GUI. Consequently, the development environment itself, as well as the programs written in it, share a unified notion of a GUI, and Squeak looks pixel-for-pixel identical on all platforms. (In case it isn't clear, the entirety of the Squeak environment is itself implemented in Squeak Smalltalk.) The upside to this is that porting the environment is simplified, and Squeak now runs on a wide range of platforms, including the Macintosh (where it was developed), Windows, various Unix platforms, and even a few PDAs. The downside is that Squeak can look, well, somewhat ugly and unsophisticated, and menus and windows in Squeak don't behave exactly the way a Macintosh user would expect. The next degree of openness of Squeak stems from its implementation: the Squeak interpreter is itself written in Smalltalk. As the story goes, the developers of Squeak, naturally, wanted to do all of their development in

Smalltalk, but writing an interpreter on top of an interpreter would be unacceptably slow. So, they wrote their Smalltalk interpreter in a subset of Smalltalk, and they wrote a Smalltalk-to-C translator, also in Smalltalk. Running the translator against the source of their interpreter gave them, finally, an independent Smalltalk interpreter, written in C, but without requiring them to ever actually program in C. (Whew!) In the end, this “translation” gave their interpreter a 450-fold speed increase. You'll want to read the full story on how Squeak was bootstrapped, and on how its runtime achieves high performance, in *Back to the Future: The Story of Squeak*; it's actually quite clever and interesting, and a testament to what can be achieved by sticking to a good idea. The importance of this is that a competent Squeak programmer is automatically qualified to play around with the internals of the Squeak environment. Contrast this to most other languages: a Perl programmer needs to know C in order to modify the Perl interpreter, and users of compiled languages have even less access to the internals of their environment. The Squeak community as a whole is thus uniquely equipped to drive its own evolution.

### Back to the Future: The Story of Squeak

<[http://users.ipa.net/~dwighth/squeak/oopsla\\_squeak.html](http://users.ipa.net/~dwighth/squeak/oopsla_squeak.html)>

For an overview of Squeak and what makes it unique, start with the Squeak Swiki. (A “wiki” is a web site where readers can contribute by adding to, and editing, the actual content; a “swiki” is a wiki implemented in Squeak.). Of particular interest are the sections *Place in the Universe* and *The History of Squeak*, as well as the FAQ.

### Squeak Swiki

<<http://minnow.cc.gatech.edu/squeak/1>>

### Place in the Universe

<<http://minnow.cc.gatech.edu/squeak/158>>

### The History of Squeak

<<http://minnow.cc.gatech.edu/squeak/389>>

### Squeak FAQ

<<http://minnow.cc.gatech.edu/squeak/471>>

There are several tutorials available as well, either specifically aimed at Squeak, or written using another Smalltalk implementation but still mostly relevant. You may want to try them all as you get up and running.

### Squeak Tutorial

<[http://squeak.cs.uiuc.edu/cphoenix\\_tutorial/intro.html](http://squeak.cs.uiuc.edu/cphoenix_tutorial/intro.html)>

### Making Smalltalk: Spreading the OO Fun

<<http://www.linuxgazette.com/issue59/steffler.html>>

### Basic Aspects of Squeak and the Smalltalk-80 Programming Language

<<http://kaka.cosc.canterbury.ac.nz/~wolfgang/cosc205/smalltalk1.html>>

### An IBM-SMALLTALK centric Tutorial

<<http://kaka.cosc.canterbury.ac.nz/~wolfgang/cosc205/STutorial/index.html>>

***"Without a doubt, the Premiere Resource Editor for the Mac OS ... A wealth of time-saving tools."***

*– MacUser Magazine Eddy Awards*

***"A distinct improvement over Apple's ResEdit."***

*– MacTech Magazine*

***"Every Mac OS developer should own a copy of Resorcerer."***

*– Leonard Rosenthol, Aladdin Systems*

***"Without Resorcerer, our localization efforts would look like a Tower of Babel. Don't do product without it!"***

*– Greg Galanos, CEO and President, Metrowerks*

***"Resorcerer's data template system is amazing."***

*– Bill Goodman, author of Smaller Installer and Compact Pro*

***"Resorcerer Rocks! Buy it, you will NOT regret it."***

*– Joe Zobkiw, author of A Fragment of Your Imagination*

***"Resorcerer will pay for itself many times over in saved time and effort."***

*– MacUser review*

***"The template that disassembles 'PICT's is awesome!"***

*– Bill Steinberg, author of Pyro! and PBTools*

***"Resorcerer proved indispensable in its own creation!"***

*– Doug McKenna, author of Resorcerer*



# Resorcerer® 2

**Version 2.0**

**The Resource Editor for the Mac™ OS Wizard**

## ORDERING INFO

Requires System 7.0 or greater,  
1.5MB RAM, CD-ROM

Standard price: \$256 (decimal)

**Website price: \$128 - \$256**

(Educational, quantity, or  
other discounts available)

Includes: Electronic documentation  
60-day Money-Back Guarantee  
Domestic standard shipping

Payment: Check, PO's, or Visa/MC  
Taxes: Colorado customers only

Extras (call, fax, or email us):  
COD, FedEx, UPS Blue/Red,  
International Shipping

MATHEMAESTHETICS, INC.  
PO Box 298  
Boulder, CO 80306-0298 USA  
Phone: (303) 440-0707  
Fax: (303) 440-0504  
resorcerer@mathemaesthetics.com

**New  
in  
2.0:**

- Very fast, HFS browser for viewing file tree of all volumes
- Extensibility for new Resorcerer Apprentices (CFM plug-ins)
- New AppleScript Dictionary ('aete') Apprentice Editor
- MacOS 8 Appearance Manager-savvy Control Editor
- PowerPlant text traits and menu command support
- Complete AIFF sound file disassembly template
- Big-, little-, and even mixed-endian template parsing
- Auto-backup during file saves; folder attribute editing
- Ships with PowerPC native, fat, and 68K versions

- Fully supported; it's easier, faster, and more productive than ResEdit
- Safer memory-based, not disk-file-based, design and operation
- All file information and common commands in one easy-to-use window
- Compares resource files, and even **edits your data forks** as well
- Visible, accumulating, editable scrap
- Searches and opens/marks/selects resources by text content
- Makes global resource ID or type changes easily and safely
- Builds resource files from simple Rez-like scripts
- Most editors DeRez directly to the clipboard
- All graphic editors support screen-copying or partial screen-copying
- Hot-linking Value Converter for editing 32 bits in a dozen formats
- Its own 32-bit List Mgr can open and edit very large data structures
- Templates can pre- and post-process any arbitrary data structure
- Includes nearly 200 templates for common system resources
- TEMPLs for Installer, MacApp, QT, Balloons, AppleEvent, GX, etc.
- Full integrated support for editing color dialogs and menus
- Try out balloons, 'ictb's, lists and popups, even create C source code
- Integrated single-window Hex/Code Editor, with patching, searching
- Editors for cursors, versions, pictures, bundles, and lots more
- Relied on by thousands of Macintosh developers around the world

To order by credit card, or to get the latest news, bug fixes, updates, and apprentices, visit our website...

**www.mathemaesthetics.com**

For further references on Squeak, the Phaidros Software web site has collected together several pieces of documentation into an online book, and you can find links to further references in the Squeak section of the Open Directory or on the *Learning to Squeak* page. Also of interest will be the *Squeak Smalltalk Quick Reference*, which will come in handy as you begin to program. Finally, for an interesting application take a look at the Squeak Web Server, an http server implemented entirely within Squeak.

#### Squeak Online Book Contents

<<http://www.phaidros.de/DIGITALIS/englisch/sqk/sqk00002.htm>>

#### Open Directory: Squeak

<<http://dmoz.org/Computers/Programming/Languages/Smalltalk/Squeak/>>

#### Learning to Squeak

<<http://minnow.cc.gatech.edu/squeak/377>>

#### Squeak Smalltalk: A Quick Reference

<<http://www.mucom.com/squeak-qref.html>>

#### Squeak Web Server

<<http://macos.tuwien.ac.at/Squeak/webServer.html>>

### THE MODEL-VIEW-CONTROLLER PATTERN

The UI of Squeak leads us to another recurring topic in Smalltalk — design patterns. Once again, Smalltalk's pure object orientation make it the natural forum for the discussion of design topics. One of the earliest design patterns, and possibly the most famous, is the Model-View-Controller (MVC) pattern, which originated as the paradigm for the user interface of Smalltalk programs. In brief, the MVC pattern is a strategy for separating the user interface of an application from its core logic, in part to promote code reuse and to insulate the different logical layers of an application from changes in other layers. In MVC, the Model is an object which contains the "business logic", devoid of any concept of how it might be displayed. The View implements the display of information — the actual windows and buttons of the user interface. Finally, the Controller handles user input (such as keyboard presses and mouse clicks) and triggers state changes in the Model and the View; it is where much of the code which is specific to a single application will live, and hence it tends to be the least likely to be reused in other applications. There are numerous variations on MVC, tailoring it to better fit different situations. Apple's *Application Architecture* documentation discusses it in connection with application design within the Cocoa frameworks. Java's Swing frameworks are also based on a variation of MVC. (See Sun's web site for more on this, in an article entitled *A Swing Architecture Overview*.) In truth, MVC is probably best thought of as an application of several different design patterns, and although it seems straightforward on the surface it becomes more complex and subtle as you think about it more deeply and begin to apply it in different situations, and it can serve as a practical, if not simple, introduction to the patterns literature. Squeak in particular is moving toward a different model for its GUI,

called Morphic, and Squeak users will want to learn about this as well, as an alternative to MVC.

#### Programming Topics: Application Architecture

<<http://developer.apple.com/techpubs/macosx/Cocoa/TasksAndConcepts/ProgrammingTopics/AppArchitecture/index.html>>

#### A Swing Architecture Overview

<<http://java.sun.com/products/jfc/tsc/articles/architecture/index.html>>

#### Morphic - The Squeak User Interface

<<http://webs.sinectis.com.ar/jmvuletich/MorphicTutorial/Morphic.html>>

For a brief explanation of MVC and its terminology, as well as some of its shortcomings, check out Object Arts' page on the subject, and for alternative high-level descriptions see CERN's web site and Michael Callaghan's notes. If you want to delve deeper into MVC, the foundational paper *How to use Model-View-Controller* provides a thorough treatment. Finally, IBM's web site has an extensive example of the pattern's application in a Java program (which, to continue our theme, is informative even if you do not program in Java).

#### Object Arts: Model-View-Controller framework

<<http://www.object-arts.com/EducationCentre/Overviews/MVC.htm>>

#### CERN: The Model-View-Controller Architecture

<<http://rd13doc.cern.ch/Atlas/Notes/004/Note004-7.html>>

#### Michael Callaghan: Model-View-Controller Architecture

<<http://www.cms.dmu.ac.uk/~jmc/mvc.html>>

#### How to use Model-View-Controller (MVC)

<<http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>>

#### Applying the Model-View-Controller Design Paradigm in VisualAge for Java

<<http://www7.software.ibm.com/vad.nsf/Data/Document2672>>

If you discover that Smalltalk's development environment is just what you have been looking for, you should take a look at ActiveDeveloper, which provides a similar environment for developing Cocoa applications, with a focus on rapid prototyping, debugging, and API exploration. Currently they do not have a version announced for the new Mac OS X, but if we are lucky they have one under development.

#### ActiveDeveloper

<<http://www.interactive-technology.com/Products/ActiveDeveloper.html>>

Enjoy your foray into Smalltalk development. It is a fascinating language, with recurring ties to Apple and the Macintosh — from the origins of the GUI at Xerox PARC, to Objective-C, to Squeak. There are many languages and development environments which are pleasant to work with, but the very culture of Smalltalk seems to be focused on fun and experimentation — it is unique and was designed from the start to genuinely be a joy to use, without compromising functionality or purity of design. 

# MacAddict's

## YOU KNOW WHO YOU ARE

**Call 1-888-771-6222**  
**TO GET YOUR FREE ISSUE AND**  
**FREE CD-ROM NOW!**

**TO RECEIVE YOUR  
FREE ISSUE AND  
FREE CD-ROM OF**

**MacAddict**

**AT NO RISK!**

If you're pleased with what you see, we'll send you 11 more issues and 11 more CD-ROMs (12 in all) for only \$29.95. That's a 68% savings off the newsstand rate.

Offer good in U.S. only. Cover price is \$7.99. Canada: U.S. \$45, includes GST. Foreign: U.S. \$58. Prepaid in U.S. funds only. Please allow 6-8 weeks for delivery of first issue.

APPLE REVS: DV POWERBOOKS! GRAPHITE iBOOKS! AND FASTER G4S!

A better machine. A better magazine.

# MacAddict

MAY/00

The **ultimate mac product slugfest!**

**31 PRODUCTS BATTLE IT OUT!**

Quark vs. InDesign  
Illustrator vs. Freehand  
GoLive vs. Dreamweaver  
Palm vs. Handspring  
Quake vs. Unreal  
DSL vs. Cable  
**AND MANY MORE!**

**9 USES FOR YOUR OLD**

**software slugfest!**

**You Pick the Winner!**

**INCLUDING:**

- Ultimate Tournament and Quake III
- Illustrator and Freehand
- GoLive and Dreamweaver
- Fudra and Outlook Express
- Audio and SoundJam MP and QDesign MVP
- And More!

**MacAddict 45**

**Dangerous Demos**

- Adobe Creative Suite
- iMovie 3.0 demo
- iPhoto 2.0 demo
- iWeb 1.0 demo
- iWork 2.0 demo
- iMovie 3.0 demo
- iPhoto 2.0 demo
- iWeb 1.0 demo
- iWork 2.0 demo

**Way Cool Ware**

- iMovie 3.0
- iPhoto 2.0
- iWeb 1.0
- iWork 2.0
- iMovie 3.0
- iPhoto 2.0
- iWeb 1.0
- iWork 2.0

MAY 2000

## List of Advertisers

|                                  |         |
|----------------------------------|---------|
| 4D .....                         | 55      |
| Active Concepts .....            | 64      |
| AD Software .....                | 19      |
| AEC Software .....               | 5       |
| Aladdin Knowledge Systems .....  | 37      |
| Aladdin Systems .....            | 63      |
| Blue World .....                 | 9       |
| Bowers Development .....         | 10      |
| Cafederic .....                  | 24      |
| Carnation .....                  | 27      |
| Catalog Stuff .....              | 66      |
| Developer Depot .....            | 28-29   |
| Digital Forest .....             | 33      |
| Drive Savers .....               | 16      |
| edition.net .....                | 31      |
| eSellerate .....                 | 45      |
| Excel Software .....             | 50      |
| FairCom Corporation .....        | 21      |
| Farallon .....                   | 73      |
| Felt Tip .....                   | 23      |
| Intego .....                     | 35      |
| Mac OS Radio .....               | 40      |
| Mac Addict .....                 | 79      |
| Mac Show .....                   | 44      |
| Main Event .....                 | 48      |
| Marathon .....                   | 8       |
| Mathemaestics .....              | 77      |
| MCE .....                        | 38      |
| Metrowerks .....                 | BC      |
| Mindvision .....                 | 17      |
| MYOB .....                       | 41      |
| Netopia .....                    | 25      |
| Onyx Technologies .....          | 11      |
| OpenBase .....                   | 43      |
| Paradigma .....                  | 26      |
| Peachpit Press .....             | 49      |
| PowerMax .....                   | 47      |
| Real Software .....              | 51      |
| Register .....                   | 65      |
| Scientific Placement .....       | 22      |
| SharpNet Solutions .....         | 13      |
| Small Dog Electronics .....      | 1       |
| Stone Tablet .....               | 20      |
| SuSE .....                       | IBC     |
| Technosoft .....                 | 56      |
| Tera Global Communications ..... | IFC, 57 |
| Terra Soft Solutions .....       | 39      |
| Thinking Home .....              | 32      |
| Thursby .....                    | 53      |
| True Basic .....                 | 15      |
| UNI SOFTWARE PLUS .....          | 6       |
| Vertical Solutions .....         | 18      |
| WIBU Systems .....               | 54      |

### Mac OS X Porting and Development Showcase

|                         |    |
|-------------------------|----|
| Red Rock Software ..... | 68 |
| Shadetree .....         | 69 |
| Omni Group, The .....   | 70 |
| Robosoft .....          | 71 |

## List of Products

|                                                                       |         |
|-----------------------------------------------------------------------|---------|
| 4D and WebSTAR • 4D .....                                             | 55      |
| AppMaker • Bowers Development .....                                   | 10      |
| BasePlant • Cafederic .....                                           | 24      |
| CodeWarrior • Metrowerks .....                                        | BC      |
| Contract Developer • Vertical Solutions .....                         | 18      |
| c-tree Plus • FairCom Corporation .....                               | 21      |
| Data Recovery • Drive Savers .....                                    | 16      |
| Dave • Thursby .....                                                  | 53      |
| Development Tools • Developer Depot .....                             | 28-29   |
| Domain Name Registration • Register.com .....                         | 65      |
| Equipment • PowerMax .....                                            | 47      |
| Equipment • Small Dog Electronics .....                               | 1       |
| FastTrack Schedule • AEC .....                                        | 5       |
| Funnel Web 3 • Active Concepts .....                                  | 64      |
| Installer Maker • Aladdin Systems .....                               | 63      |
| Internet Service Provider • Digital Forest .....                      | 33      |
| Job Placement • Scientific Placement .....                            | 22      |
| Lasso Web Data Engine • Blueworld .....                               | 9       |
| Linux • SuSE .....                                                    | IBC     |
| Macintosh terminal emulation • Carnation Software .....               | 27      |
| MacHasp USB • Aladdin Knowledge Systems .....                         | 37      |
| NetBarrier • Intego .....                                             | 35      |
| NetOctopus • Netopia .....                                            | 25      |
| OO File • AD Software .....                                           | 19      |
| PowerBook Products • MCE .....                                        | 38      |
| Professional Macintosh and Internet Development • Thinking Home ..... | 32      |
| Publication • MacAddict .....                                         | 79      |
| Publications • Peach Pit Press .....                                  | 49      |
| Rackmount solutions and accessories • Marathon Computer .....         | 8       |
| RAD Studio • OpenBase International .....                             | 43      |
| Radio Show • Mac OS Radio .....                                       | 40      |
| Radio Show • Mac Show Live .....                                      | 44      |
| REALbasic • REAL Software .....                                       | 51      |
| Resorcerer • Mathmaesthetics .....                                    | 77      |
| Scripter • Main Event .....                                           | 48      |
| SkyLINE 11MB • Farallon .....                                         | 73      |
| Sound Studio • Felt Tip Software .....                                | 23      |
| Spotlight • Onyx Technologies .....                                   | 11      |
| Stone Table • Stone Table Publishing .....                            | 20      |
| SQL Database Design in MacA&D • Excel Software .....                  | 50      |
| Tera Media • Tera Global Communications .....                         | IFC, 57 |
| True Basic • TrueBasic .....                                          | 15      |
| USB Stuff, FireWire Stuff • Catalog Stuff .....                       | 66      |
| Valentina • Paradigma .....                                           | 26      |
| WISE • Mindvision .....                                               | 17      |
| VOODOO Server • UNI SOFTWARE PLUS .....                               | 6       |
| Web Hosting • edition.net .....                                       | 31      |
| WIBU-KEY Software Protection • WIBU SYSTEMS AG .....                  | 54      |
| Yellow Dog Linux • Terra Soft Solutions .....                         | 39      |

### Mac OS X Porting and Development Showcase

|                                                    |    |
|----------------------------------------------------|----|
| Software Development Outsourcing • Shadetree ..... | 69 |
| OS X Development • Red Rock Software .....         | 68 |
| Consultants • The Omni Group .....                 | 70 |
| Porting and Implementation • Robosoft .....        | 71 |

The index on this page is provided as a service to our readers. The publisher does not assume any liability for errors or omissions.

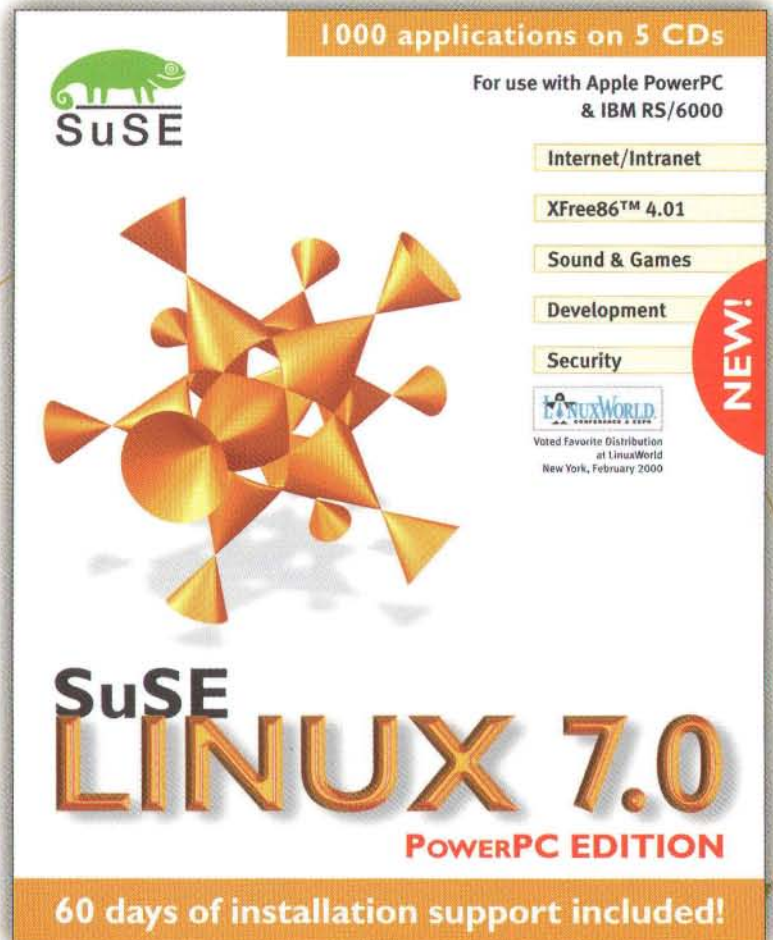
# SuSE Linux gives you all the **POWER** — without the pricetag.

For PowerPC and RS/6000, you get:

**POWERFUL** graphics manipulation:  
Use the GIMP (GNU Image Manipulation Program) to do all your imaging tasks — even video editing.

A **POWERFUL** development environment: C/C++, Fortran, Pascal, Java, Perl, Python...

A **POWERFUL** server platform:  
Full support for AppleTalk, TCP/IP, IPSec, NFS, UUCP, SLIP, PPP, IPX, ISDN, Samba, TokenRing, ARCnet, Internet.



\$49.95

60 days installation support • 1000 open-source applications • 5 CD-ROMS



The freedom to change.  
The power to change the Linux world!

[www.SuSE.com](http://www.SuSE.com) 510.628.3380

Available today at major software retailers and on [www.suse.com](http://www.suse.com).

**FREE**  
Online Course!  
**Intro to**  
**PowerPlant**  
enroll at:  
[www.CodeWarriorU.com](http://www.CodeWarriorU.com)



# Keeping Mac dreams alive since 1993.

Two great operating systems,  
one CodeWarrior.

**M**ac developers depended on CodeWarrior to make the platform architecture shift from 68K to PowerPC processors. Now CodeWarrior does it again, speeding your transition to the next new operating system. CodeWarrior for Mac OS, Version 6.0 supports development for both OS X

and Classic Mac operating systems from a single, powerful, award-winning Integrated Development Environment. Discover how CodeWarrior for Mac OS, Version 6.0 can help you realize your Mac development dreams. Visit [www.metrowerks.com/go/mac](http://www.metrowerks.com/go/mac).

# CodeWarrior®

  
**metrowerks®**  
Software Starts Here